

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/366692337>

# From Emotion AI to Cognitive AI

Article · December 2022

DOI: 10.53941/ijndi0101006

---

CITATIONS

5

READS

463

3 authors:



Guoying Zhao

University of Oulu

360 PUBLICATIONS 19,235 CITATIONS

[SEE PROFILE](#)



Yante Li

University of Oulu

18 PUBLICATIONS 202 CITATIONS

[SEE PROFILE](#)



Qianru Xu

University of Oulu

26 PUBLICATIONS 154 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Group affect analysis [View project](#)



texture [View project](#)

---

Survey/review study

# From Emotion AI to Cognitive AI

Guoying Zhao\*, Yante Li, and Qianru Xu

University of Oulu, Pentti Kaiteran Katu 1, Linnanmaa 90570, Finland

\* Correspondence: [guoying.zhao@oulu.fi](mailto:guoying.zhao@oulu.fi)

Received: 22 September 2022

Accepted: 28 November 2022

Published: 22 December 2022

**Abstract:** Cognitive computing is recognized as the next era of computing. In order to make hardware and software systems more human-like, emotion artificial intelligence (AI) and cognitive AI which simulate human intelligence are the core of real AI. The current boom of sentiment analysis and affective computing in computer science gives rise to the rapid development of emotion AI. However, the research of cognitive AI has just started in the past few years. In this visionary paper, we briefly review the current development in emotion AI, introduce the concept of cognitive AI, and propose the envisioned future of cognitive AI, which intends to let computers think, reason, and make decisions in similar ways that humans do. The important aspect of cognitive AI in terms of engagement, regulation, decision making, and discovery are further discussed. Finally, we propose important directions for constructing future cognitive AI, including data and knowledge mining, multi-modal AI explainability, hybrid AI, and potential ethical challenges.

**Keywords:** emotion; cognition; human intelligence; artificial intelligence

---

## 1. Introduction

Intelligence is what makes us human [1]. Human intelligence is a mental attribute that consists of the ability to learn from experience, adapt to new conditions, manage complex abstract concepts, and interact with and change the environment using knowledge. It is the human intelligence that makes Humans different from other creatures.

Emotional and cognitive functions are inseparable from the human brain and they jointly form human intelligence [2, 3]. Emotions are biological states associated with the nervous system. They are brought on by neurophysiological changes associated with thoughts, feelings, behavioral responses, and pleasure or displeasure. Humans express their feelings or react to external and internal stimuli through emotion. Thus, emotion plays an important role in everyday life. In recent years, research on emotion has increased significantly in different fields such as psychology, neuroscience, and computer science, where the so-called emotion artificial intelligence (AI) is committed to developing systems to recognize, process, interpret and simulate human emotion, leading to various potential applications in human lives including E-teaching, home care, security, and so on.

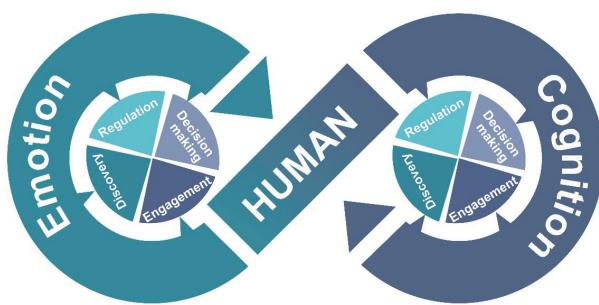
Cognition refers to the mental action or process related to acquiring knowledge and understanding through thought, experience, and sense [4–7]. Through cognitive processing, we make decisions and produce appropriate responses. Research shows that cognition evolved hand-in-hand with emotion [2, 3]. In other words, the emotion and cognition are two sides of the same system, and separating them leads to anomalous behaviors. However, given the importance of cognition to human intelligence, research on cognitive AI is still very limited. To better understand human needs and advance the development of AI, it is crucial to build computational models with cognition functions to understand human thoughts and behaviors and import the cognition capability to machines for better interaction, communication, and cooperation in human-computer interactions or computer-assisted human-human interactions. In the future, modeling human cognition and reconstructing it with emotion in real-world application systems are the new objectives of AI [8].

By their nature, as both emotion AI and cognitive AI are inspired and built to imitate human intelligence, they share a lot in common and require the combined efforts of multiple disciplines. In this sense, the focus of this paper is to discuss the future cognitive AI, which makes computers capable of analyzing, reasoning, and making decisions like human users. We firstly introduce the relationship between the emotion and cognition. Then, the emotion AI and cognitive AI are introduced. The important aspects of cognitive AI are specifically discussed including engagement, regulation, decision making, and discovery. Finally, we share the opinions of research focus for future cognitive AI.

## 2. Emotion and Cognition

### 2.1. Emotion-Cognition Interactions

Cognition and emotion appear to be separate from each other, but actually, they are two sides of the same coin, as shown in **Figure 1**. In fact, for a long historical time dating back to the time of Plato, emotion and cognition have been considered as separate and independent processes [9, 10]. In recent years, however, more and more studies tend to consider emotion and cognition as interrelated and integrated [11]. Numerous psychological studies have found that the processing of salient emotional stimuli and the experience of affective states can have an impact on behavioral and cognitive processing, while in turn cognition can influence and regulate emotion [12, 13]. From a neurobiological perspective, in the human brain, although previous studies have tended to suggest that emotion is responsible for the subcortical processing such as the primitive limbic system, while cognitive-related processes are mostly processed by the cortex areas like the prefrontal cortex, substantial studies in recent years have shown that emotional and cognitive processes are interrelated with each other and shared a large number of overlapping regions [14, 15], for reviews, please see [12, 16]. Therefore, the study of the interaction between emotion and cognition can help us create more trustworthy and human-like AI and enhance human-computer interactions.



**Figure 1.** Emotion and cognition.

### 2.2. Emotion AI

Emotion plays an essential role in human communication and cognition. With emotions, humans can express their various feelings or react accordingly to internal and external stimuli. While not all AI or intelligent software systems need to be emotional or have exactly all human-like emotions such as depression and anxiety, using AI to further understand emotions and develop corresponding systems would certainly make AI more user-friendly and more convenient for our daily lives [17].

For this purpose, emotion AI is developed and dedicated to uncovering human emotions and enabling computers to have human-like capabilities to understand, interpret, and even express emotions [18]. Emotion AI can be used in a wide range of areas where emotions play an important role [17]. For instance, in mental health care, AI may help psychologists to detect symptoms or disorders that are hidden or unaware of by patients. In the education field, an AI teacher who can understand students' emotional states and respond appropriately would enhance the learning process, and thus is utilized in remote or hybrid learning scenarios. Furthermore, emotion AI is coming into use in many other fields such as entertainment, communication, and intelligent agents [17]. To date, the novel methodology developed in affective computing and Emotion AI has significantly contributed to our lives in many areas such as emotional well-being, E-teaching, home care, and security.

Currently, there are three mainstreams in artificial emotional intelligence, namely emotion recognition, emotion synthesis, and emotion augmentation [19]. Emotion recognition refers to applying computer systems to recognize human emotions in affective computing, and is one of the most traditional approaches that has been employed in emotion AI research [20]. Emotion recognition has been widely used in different modalities such as facial expressions [21–23], body gestures [24, 25], acoustic or written linguistic content [26, 27], physiological signals [28, 29], and the fusion of multiple modalities [30–32]. While most previous research has focused on facial expression recognition, given the multi-modal nature of emotions, more and more researchers are tending to refer to other modalities as well.

With these multi-modal cues, AI can better recognize genuine and spontaneous emotions and build more reliable systems accordingly. Besides recognizing human emotions, AI often needs to analyze and process human behaviors and interact with humans, so it is important to equip AI with emotion synthesis capability [33] to improve the human-computer interactive experience. The main development of emotion synthesis is in speech, facial expression, and body behavior synthesis, which can even date back to three decades ago [19]. Exploring emotion synthesis, undoubtedly, can enhance the AI's affective and social realism, thus improving the reliability of AI and making their

interaction with users more natural [34]. However, it is important to note the emotion synthesis we raise here is far from an artificially emotional machine, as emotion synthesis is more about external emotional expressions, i. e., equipping AI with some human-like emotional reactions. There is still a long way to go to let AI generate inner emotions, which involves a lot of ethical issues that cannot be ignored.

In addition, emotion augmentation denotes embedding the principle of emotion in AI and using it in planning, reasoning, or more general goal achievement [19, 35]. In other words, emotion augmentation requires applying emotion concepts to achieve broader goals in AIs. For instance, by introducing two emotional parameters, namely anxiety and confidence, the emotional backpropagation (BP) learning algorithm can reach an outstanding performance compared to the conventional BP-based neural network in the facial recognition task [36]. Apart from that, emotion augmentation can be also used in a wider range of scenarios, such as designing models, computational reinforcement learning, or other cognitive-related tasks [35]. Although still far away, the booming development of studies in sentiment analysis and affective computing (in computer science and other interdisciplinary disciplines) makes the goal of translating emotions into AI less far-fetched than one might think.

### 2.3. Cognitive AI

Human thinking is beyond imagination. What human think about every day involves: how to act in a dynamic and evolving world; how to juggle multiple goals and preferences; how to face opportunities and threats, etc. Is it possible that a computer develops the ability to think and reason without human intervention? An overarching goal for future AI is to simulate human thought processes in a computerized model. The result is cognitive AI.

Cognitive AI is derived from AI and cognitive science [37]. It develops computer systems simulating human brain functions [38]. The aim of cognitive AI is to mimic human behaviors and solve complex problems. It encompasses the main brain behaviors of human intelligence, including perception, attention, thought, memory, knowledge formation, judgment and evaluation, problem-solving and decision making, etc [39]. Moreover, cognitive processes utilize existing knowledge to discover new knowledge.

By its nature, cognitive AI needs to employ methods from multiple disciplines such as cognitive science, psychology, linguistics, physics, information theory, and mathematics [40, 41]. Based on the knowledge from different disciplines, cognitive AI uses computer theory and techniques to simulate human cognitive tasks in order to make computers think like the human brain [41]. Cognitive AI is fundamentally different from traditional AI. Traditional AI, or traditional computer techniques, are mainly based on the fed data and processed accordingly under human pre-programming [42]. On the contrary, cognitive AI should be able to continuously adapt itself to the context and environment so as to evolve and grow over time. For example, IBM Watson, a typical representative of Cognitive AI, defeated the human champion in the Jeopardy game in 2011 [43]. Unlike its forerunner Deep Blue (known as the first computer that defeated a human chess player), which required an exhaustive search and performed merely quantitative analysis, Watson relies on offline stored information and uses natural language processing combined with appropriate contextual information to perform adaptive reasoning and learning to produce answers [41, 43].

In this sense, the benefits of cognitive AI outweigh traditional AI, as traditional AI prefers to execute optimal algorithms to solve problems, while cognitive AI goes a big step further to replicate human wisdom and intelligence through multifaceted analysis including understanding and modeling human engagement, regulation (self-regulation and shared regulation in interactions), decision making, discovery, etc. Notice that cognitive AI requires complex data learning, knowledge mining, multi-model explainability, and hybrid AI to learn and analyze the patterns and to ensure that the reasoning results are reliable and consistent [44].

## 3. Important Aspects in Cognitive AI

In this section, we specifically discuss the important aspects of developing a successful cognitive AI system, including engagement, regulation, decision making, and discovery.

### 3.1. Engagement

Engagement describes the effortful commitment to goals [45]. Cognition, emotion, and motivation are incorporated into the engagement. Multiple studies show that positive emotion and motivation can improve the engagement of humans [46]. These findings have been successfully utilized in various fields, such as education, business dealing and human-computer interaction (HCI) to improve the engagement of users [47–49].

In recent years, besides emotion and motivation, researchers start to study the contribution of cognitive AI to engagement. IBM developed Watson taking advantage of cognitive AI for customer engagement to better identify the wants and needs of the customers [50]. In healthcare, cognitive computing systems collect individual health data from a variety of sources to enhance patient engagement and help professionals treat patients in a customized manner [51]. However, the technology is still at an early stage. How to find what really matters in engaging humans requires fur-

ther study.

### 3.2. Regulation

In social cognitive theory, human behavior is motivated and regulated through constant self-influence. The self-regulative mechanism has a strong impact on affect, thought, motivation, and human action [52]. Regulation involves the ability to make adaptive changes in terms of cognition, motivation, and emotions in challenging situations [53]. Emotion changes could be the reason for the regulation, while they also could be the result of regulation [54].

In Bandura's theory [55], the regulation process contains self-observation, judgment, and self-response. The self-observation is how humans look at themselves and their behaviors. Judgment refers to using a standard to compare what humans see. Self-response is performing different behaviors according to whether meeting the standard or not. Through these steps, humans are able to control their behavior [52]. However, until now, it is almost impossible to evidence the "invisible mental" shared regulation due to methodological limitations [56]. Moreover, it is not clear how humans regulate their internal states in response to internal and external stressors [52]. In the future, the self- and shared-regulative mechanisms should be further investigated and integrated into the AI systems to make AI learn by itself like humans.

### 3.3. Decision Making

Decision making is regarded as the cognitive process referring to problem-solving activities yielding a solution deemed to be optimal among many alternatives [57]. The decision making process can be based on explicit or tacit knowledge and beliefs. It can be rational or irrational.

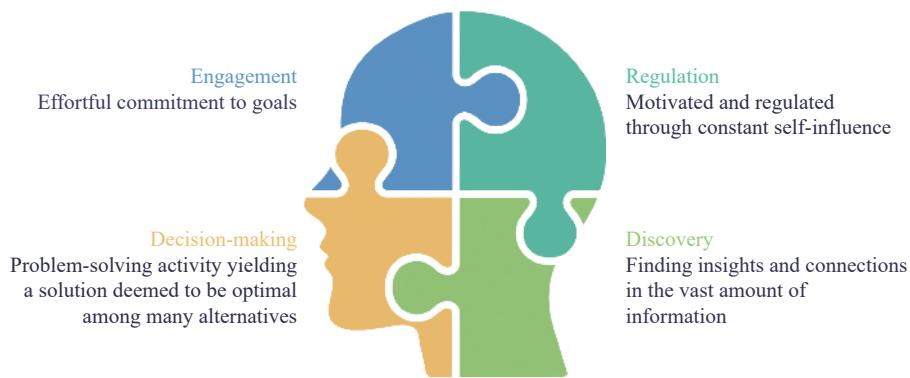
Decision making happens every now and then in human's daily life. A wise decision may make life and the world better. However, human's decisions may be influenced by multiple factors, especially emotion. Human choices can be influenced by the human's mood at the time of decision making [58, 59]. For instance, happy people are more likely to be optimistic about the current situation and choose to maintain the current state, whereas sad people are more inclined to be pessimistic about the current situation and choose to change the current state. Fearful people tend to hold pessimistic judgments about future events, while angry people tend to keep optimistic judgments about future events. It is important to study how to model the connections between emotion and decision making and understand the deep process of decision making in human brains.

### 3.4. Discovery

As a high-level and advanced scope of cognitive AI, discovery refers to finding insights and connections in the vast amount of information and developing skills to extract new knowledge [60]. With the increasing volumes of data, it becomes possible to train models with AI technologies to efficiently discover useful information from massive data. At the same time, the massive complex data increases the difficulty of manually processing by humans. Therefore, AI systems are needed to help exploit information more effectively than humans could do on their own.

In recent years, some systems with discovery capabilities have already emerged. The cognitive information management shell at Louisiana State University is a cognitive solution and has been applied in preventing future oil spills [61]. More specifically, this system built a complex event processing system that can detect problems in mission-critical and generate intelligent decisions to modulate the system environment. The system contains distributed intelligent agents to collect multiple streaming data to build interactive sensing, inspection, and visualization systems. It is able to achieve real-time monitoring and analysis. Through archiving past events and cross-referencing them with current events, the system can discover deeply hidden patterns and then make decisions upon them. Specifically, cognitive information management not only sends alerts but also dynamically re-configures to isolate critical events and repair failures. Currently, the study of system with discovery capabilities are still in the early stages. The value propositions for the future are very compelling. In the future, we should consider more about how to design cognitive systems with discovery ability effectively and efficiently so as to solve different tasks in complex real-life situations.

In general, engagement, self-regulation, decision making, and discovery are considered to be complementary in cognitive AI systems, as shown in [Figure 2](#). Engagement refers to the way how humans and systems interact with each other. Through enhancing the engagement, more effort would be contributed to achieving the final goal. In this process, adaptions referring to regulation are ongoing which further simulates the engagement [62]. Furthermore, engagement and regulation can influence the ability to explore the underlying complex relationships and solve problems that are related to discovery and decision-making, relatively [63]. In turn, just like us human beings, discovery and decision-making can also impact engagement and regulation. To develop a successful cognitive AI system, all of the above four aspects should be considered and well-designed.



**Figure 2.** Four aspects of cognitive AI are complementary to each other.

#### 4. The Future of Cognitive AI

Towards cognitive AI, data and knowledge mining, multi-modal AI explainability, and hybrid AI should be explored, and the potential new ethical issues should also be kept in mind.

##### 4.1. Data and Knowledge Mining

Fueled by advances in AI and the boosting of digital data, the cognitive process could be increasingly delegated to automated processes based on data analysis and knowledge mining [64], which intends to dig hidden patterns and knowledge from vast amounts of information through AI technology [65]. The new AI systems should support analytical engagement, regulations, decision making, and discovery approaches, which involve big data of greater variety, higher volumes and more velocity with the technology development [66]. Most of the current machine learning techniques based on big data are usually based on the vector space and work for specific scenarios [67–69]. It is difficult to store, analyze, and visualize big and complex cognitive data for further processes or results with traditional statistical models. A promising future research focus is to learn from dynamic data of complicated cognitive processes in both Euclidean space and non-Euclidean space with uncertainties for generic scenarios and scenarios with equivocality [70]. Another research direction would be the life-long learning and self-supervised learning that discover new knowledge through digging into the relationships among the data with complex structures and uncertainties. For example, in the field of cybersecurity where dynamic threats and cyber-attacks appear all the time [71], cybersecurity is required to build fast self-reliant cognitive computing systems that can make wiser security decisions based on self-learned knowledge and complex surrounding factors.

##### 4.2. Multi-Modal AI Explainability

Cognitive systems are dedicated to developing advanced human-like AIs. However, as AI becomes more intelligent and impacts our lives in many ways, an insurmountable problem is how we trace back and understand the results of AI algorithms.

Explainable artificial intelligence (XAI) refers to a type of explainable AI that generate details and reasoning processing to help users understand its functions [72]. In terms of cognitive AI, it is often necessary to collect information from different modalities. For example, in building emotion and cognitive AI engines, behavioral information such as facial expressions and body gestures, language and voice information, as well as physiological signals are all important data sources for understanding human emotion and cognition functions, and are collected simultaneously across modalities. However, once we feed all data into the computer, the whole calculation process seems to be a “black box”, and even the researchers or engineers who develop the algorithms themselves cannot know what happens inside the “box” [72, 73]. Therefore, developing cognitive XAI to help us interpret the results, reveal the reasoning processing, and find causality among multi-modal data variables would be a fruitful area for further work. The study of multi-modal AI explainability could have various applications, especially in health care. On one hand, the cognitive XAI could help doctors better analyze diseases and reduce the rate of misdiagnosis [42, 74]. On the other hand, the patient can better understand their health status which leads to increased trust in treatment options.

##### 4.3. Hybrid AI

Hybrid AI is the future direction, which represents joint intelligence from human beings and algorithms. To date, hybrid AI has gained much more attention and started to be applied to some realistic applications, such as management [75]. However, most of them are simple combinations of human and AI decisions. The way to better take advantage of humans and algorithms should be explored. Moreover, in daily life, efficient interactions and collaborations along with optimal decisions require contributions from multiple people/systems who/which have various

strengths. It is valuable to study how to combine the decision of a group of people/systems. In general, when addressing complexity, AI can extend humans' cognition, while humans offer an intuitive, generic, and creative approach to deal with uncertainties and equivocality. Therefore, hybrid intelligence can strengthen both human intelligence and AI, and get both co-evolving by integrating cognitive AI into the development. The hybrid cognitive AI benefits high-risk situations, such as medical diagnostics, algorithmic trading, and autonomous driving [76].

#### 4.4. Ethical Concerns

If AI algorithms/systems have more emotional and cognitive intelligence, they would be more human-like and play a more important role in various applications, but more ethical concerns might be introduced, which cannot be overlooked in constructing future cognitive computing systems.

AI is usually driven by big data, especially nowadays many researchers emphasize the use of multi-modal data to improve the accuracy of algorithms. However, a large amount of data, especially from the same individual, will also introduce privacy and potential misuse issues. There is also a growing concern about the risks of false interpretation of emotions and false cognitive actions due to inaccurate and unreliable AI algorithms. AI may reveal emotional or cognitive states that people do not want to disclose or be treated unfairly due to the bias of the AI's decisions. Therefore, it is important to consider the ethical issues and comply with the corresponding principles, e.g., transparency, justice, fairness and equity, non-maleficence, responsibility, and privacy. For a detailed guideline, please see [77] while improving the performance of AI.

### 5. Conclusion

Emotion AI and cognitive AI simulate human intelligence together. Emotion AI is dedicated to discovering human emotions and enabling computers to have human-like capabilities to understand, interpret, and synthesize emotions. Cognitive AI aims to make computers analyze, reason, and make decisions like a human. In this paper, we review the development and relationship of emotion AI and cognitive AI. Cognitive computing is recognized as the next era of computing. Four important aspects of cognitive AI are discussed including engagement, regulation, decision making, and discovery, which are complementary in cognitive AI. To develop a successful cognitive computing system, all of the above four aspects should be considered. Finally, this paper explores the future of cognitive AI. Possible research directions on future cognitive AI include data and knowledge mining, multi-modal AI explainability, hybrid AI, and potential ethical issues.

**Author Contributions:** Guoying Zhao, Yante Li and Qianru Xu: conceptualization; Yante Li and Qianru Xu: investigation; Yante Li and Qianru Xu: writing — original draft preparation; Guoying Zhao: writing — review and editing; Guoying Zhao: supervision; Guoying Zhao: project administration; Guoying Zhao: funding acquisition. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is funded by the Academy of Finland for Academy Professor project Emotion AI, grant Nos 336116 and 345122, and Ministry of Education and Culture of Finland for AI forum project.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Acknowledgments:** Thanks Guoying Zhao for figure design.

### References

1. Luckin, R. *Machine Learning and Human Intelligence: The Future of Education for the 21st Century*; UCL IOE Press, London, 27, July, 2018. doi:10.1177/14782103221117655
2. Phelps, E.A. Emotion AND cognition: Insights from studies of the human amygdala. *Annu. Rev. Psychol.*, 2006, 57: 27–53.
3. Isaacowitz, D.M.; Charles, S.T.; Carstensen, L.L. Emotion and cognition. In *The Handbook of Aging and Cognition*; Salthouse, T.A., Ed.; Lawrence Erlbaum Associates Inc.: Mahwah, 2000; pp. 593–631.
4. Jessica A.Sommerville. Social Cognition. Encyclopedia of Infant and Early Childhood Development, 2020, pp.196-206, Elsevier. doi: 10.1016/B978-0-12-809324-5.21640-4.
5. Jian, M.W.; Zhang, W.Y.; Yu, H.; et al, Saliency detection based on directional patches extraction and principal local color contrast. *J. Vis. Communun. Image Represent.*, 2018, 57: 1–11.
6. Jian, M.W.; Qi, Q.; Dong, J.Y.; et al, Integrating QDWD with pattern distinctness and local contrast for underwater saliency detection. *J. Vis. Communun. Image Represent.*, 2018, 53: 31–41.
7. Jian, M.W.; Lam, K.M.; Dong, J.Y.; et al, Visual-patch-attention-aware saliency detection. *IEEE Trans. Cybern.*, 2015, 45: 1575–1586.
8. Schmid, U, Cognition and AI. *Fachbereich 1 Künstliche Intelligenz der Gesellschaft für Informatik e.V.*, GI, 2008, 1: 5.
9. Scherer, K.R. On the nature and function of emotion: A component process approach. In *Approaches to Emotion*; Scherer, K.R.; Ekman, P., Eds.; Psychology Press: New York, 1984; p. 26.

10. Lazarus, R.S. The cognition-emotion debate: A bit of history. In *Handbook of Cognition and Emotion*; Dalgleish, T.; Power, M.J., Eds.; John Wiley & Sons: New York, **1999**; pp. 3–19.
11. Pessoa, L. *The Cognitive-Emotional Brain: From Interactions to Integration*; MIT Press: Cambridge, 2013.
12. Okon-Singer, H.; Hendler, T.; Pessoa, L.; et al. The neurobiology of emotion–cognition interactions: Fundamental questions and strategies for future research. *Front. Hum. Neurosci.*, **2015**, *9*: 58.
13. Dolcos, F.; Katsumi, Y.; Moore, M.; et al. Neural correlates of emotion–attention interactions: From perception, learning, and memory to social cognition, individual differences, and training interventions. *Neurosci. Biobehav. Rev.*, **2020**, *108*: 559–601.
14. Shackman, A.J.; Salomons, T.V.; Slagter, H.A.; et al. The integration of negative affect, pain and cognitive control in the cingulate cortex. *Nat. Rev. Neurosci.*, **2011**, *12*: 154–167.
15. Young, M.P.; Scanneil, J.W.; Burns, G.A.P.C.; et al. Analysis of connectivity: Neural systems in the cerebral cortex. *Rev. Neurosci.*, **1994**, *5*: 227–250.
16. Pessoa, L. On the relationship between emotion and cognition. *Nat. Rev. Neurosci.*, **2008**, *9*: 148–158.
17. Martinez-Miranda, J.; Aldea, A. Emotions in human and artificial intelligence. *Comput. Hum. Behav.*, **2005**, *21*: 323–341.
18. Tao, J.H.; Tan, T.N. Affective computing: A review. In *Proceedings of the 1st International Conference on Affective Computing and Intelligent Interaction, Beijing, China, October 22–24, 2005*; Springer: Beijing, China, 2005; pp. 981–995. doi:10.1007/11573548\_125
19. Schuller, D.; Schuller, B.W. The age of artificial emotional intelligence. *Computer*, **2018**, *51*: 38–46.
20. Saxena, A.; Khanna, A.; Gupta, D. Emotion recognition and detection methods: A comprehensive survey. *J. Artif. Intell. Syst.*, **2020**, *2*: 53–79.
21. Li, Y.T.; Wei, J.S.; Liu, Y.; et al. 2022, Deep learning for micro-expression recognition: A survey. *IEEE Trans. Affect. Comput.*, **2022**, *13*: 2028–2046.
22. Liu, Y.; Zhou, J.Z.; Li, X.; et al. Graph-based Facial Affect Analysis: A Review. *IEEE Trans. Affect. Comput.*, **2022**, *19*: 1–20.
23. Liu, Y.; Zhang, X.M.; Zhou, J.Z.; et al. SG-DSN: A Semantic Graph-based Dual-Stream Network for facial expression recognition. *Neurocomputing*, **2021**, *462*: 320–330.
24. Chen, H.Y.; Liu, X.; Li, X.B.; et al. Analyze spontaneous gestures for emotional stress state recognition: A micro-gesture dataset and analysis with deep learning. In *Proceedings of 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 14–18 May 2019*; IEEE: Lille, France, **2019**; pp. 1–8. doi:10.1109/FG.2019.8756513
25. Liu, X.; Shi, H.L.; Chen, H.Y.; et al. iMiGUE: An identity-free video dataset for micro-gesture understanding and emotion analysis. In *Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021*; IEEE: Nashville, USA, **2021**; pp. 10631–10642. doi:10.1109/CVPR46437.2021.01049
26. Koolagudi, S.G.; Rao, K.S. Emotion recognition from speech: A review. *Int. J. Speech Technol.*, **2012**, *15*: 99–117.
27. Zhou, Z.H.; Zhao, G.Y.; Hong, X.P.; et al. A review of recent advances in visual speech decoding. *Image Vis. Comput.*, **2014**, *32*: 590–605.
28. Yu, Z.T.; Li, X.B.; Zhao, G.Y. Facial-Video-Based Physiological Signal Measurement: Recent advances and affective applications. *IEEE Signal Process. Mag.*, **2021**, *38*: 50–58.
29. Shu, L.; Xie, J.Y.; Yang, M.Y.; et al. A review of emotion recognition using physiological signals. *Sensors*, **2018**, *18*: 2074.
30. Li, X.B.; Cheng, S.Y.; Li, Y.T.; et al. 4DME: A spontaneous 4D micro-expression dataset with multimodalities. *IEEE Trans. Affect. Comput.* **2022**, in press. doi:10.1109/TAFFC.2022.3182342
31. Huang, X.H.; Kortelainen, J.; Zhao, G.Y.; et al. Multi-modal emotion analysis from facial expressions and electroencephalogram. *Comput. Vis. Image Underst.*, **2016**, *147*: 114–124.
32. Saleem, S.M.; Abdullah, A.; Ameen, S.Y.A.; et al. Multimodal emotion recognition using deep learning. *J. Appl. Sci. Technol. Trends*, **2021**, *2*: 52–58.
33. Lisetti, C.L. Emotion synthesis: Some research directions. In *Proceedings of the Working Notes of the AAAI Fall Symposium Series on Emotional and Intelligent: The Tangled Knot of Cognition, Orlando, FL, USA, October 22–24, 1998*; AAAI Press: Menlo Park, USA, 1998; pp. 109–115.
34. Hudlicka, E. Guidelines for designing computational models of emotions. *Int. J. Synth. Emotions (IJSE)*, **2011**, *2*: 26–79.
35. Strömfelt, H.; Zhang, Y.; Schuller, B.W. 2017. Emotion-augmented machine learning: Overview of an emerging domain. In *Proceedings of 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII), San Antonio, TX, USA, 23–26 October 2017*; IEEE: San Antonio, USA, **2017**; pp. 305–312. doi:10.1109/ACII.2017.8273617
36. Khashman, A. A modified backpropagation learning algorithm with added emotional coefficients, *IEEE Trans. Neural Netw.*, **2008**, *19*: 1896–1909.
37. Hwang, K.; Chen, M. Big-Data Analytics for Cloud, *IoT and Cognitive Computing*; John Wiley & Sons: Hoboken, NJ, USA, **2017**.
38. Wang, Y.X. A cognitive informatics reference model of autonomous agent systems (AAS). *Int. J. Cognit. Inf. Nat. Intell.*, **2009**, *3*: 1–16.
39. Li, J.H.; Mei, C.L.; Xu, W.H.; et al. Concept learning via granular computing: A cognitive viewpoint. *Inf. Sci.* **2015**, *298*, 447–467. doi:10.1016/j.ins.2014.12.010
40. Chen, M.; Herrera, F.; Hwang, K. Cognitive computing: Architecture, technologies and intelligent applications. *IEEE Access*, **2018**, *6*: 19774–19783.
41. Gudivada, V.N. Cognitive computing: Concepts, architectures, systems, and applications. *Handb. Stat.*, **2016**, *35*: 3–38.
42. Sreedevi, A.G.; Harshitha, T.N.; Sugumaran, V.; et al. Application of cognitive computing in healthcare, cybersecurity, big data and IoT: A literature review. *Inf. Process. Manage.* **2022**, *59*, 102888. doi:10.1016/J.IPM.2022.102888
43. Ferrucci, D.A. Introduction to “This is Watson”. *IBM J. Res. Dev.*, **2012**, *56*: 1.1–1.5.
44. Hurwitz, J.S.; Kaufman, M.; Bowles, A. *Cognitive Computing and Big Data Analytics*; John Wiley & Sons: Indianapolis, IN, USA, **2015**.
45. Fairclough, S.H.; Venables, L. Prediction of subjective states from psychophysiology: A multivariate approach. *Biol. Psychol.*, **2006**, *71*: 100–110.
46. Teixeira, T.; Wedel, M.; Pieters, R. Emotion-induced engagement in internet video advertisements. *J. Mark. Res.*, **2012**, *49*: 144–159.
47. Skinner, E.; Pitzer, J.; Brule, H. The role of emotion in engagement, coping, and the development of motivational resilience. In *International Handbook of Emotions in Education*; Pekrun, R.; Linnenbrink-Garcia, L., Eds.; Routledge: New York, **2014**; pp. 331–347.
48. Mathur, A.; Lane, N.D.; Kawsar, F. Engagement-aware computing: Modelling user engagement from mobile contexts. In *Proceed-*

- ings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; ACM: Heidelberg, Germany, 2016;* pp. 622–633. doi:10.1145/2971648.2971760
49. Renninger, K.A.; Hidi, S.E. *The Power of Interest for Motivation and Engagement*; Routledge: New York, 2015.
  50. Klie, L, IBM's Watson brings cognitive computing to customer engagement. *Speech Technol. Mag.*, 2014, 19: 38–42.
  51. Behera, R.K.; Bala, P.K.; Dhir, A, The emerging role of cognitive computing in healthcare: A systematic literature review. *Int. J. Med. Inf.*, 2019, 129: 154–166.
  52. Bandura, A, Social cognitive theory of self-regulation. *Organ. Behav. Hum. Decis. Process.*, 1991, 50: 248–287.
  53. Schunk, D.H.; Greene, J.A. Historical, contemporary, and future perspectives on self-regulated learning and performance. In *Handbook of Self-Regulation of Learning and Performance*; Schunk, D.H.; Greene, J.A., Eds.; Routledge: New York, 2018; pp. 1–15. doi:10.4324/9781315697048-1
  54. McRae, K.; Gross, J.J. Emotion regulation. *Emotion*, 2020, 20: 1–9.
  55. Bandura, A. *Social Foundations of Thought and Action*; Prentice Hall, New York, 2002. Available online: <https://psycnet.apa.org/record/1985-98423-000>(accessed on 5 November 2022).
  56. Azevedo, R.; Gašević, D, Analyzing multimodal multichannel data about self-regulated learning with advanced learning technologies: Issues and challenges. *Comput. Hum. Behav.*, 2019, 96: 207–210.
  57. Wilson, R.A.; Keil, F.C. *The MIT Encyclopedia of the Cognitive Sciences*; A Bradford Book, West Yorkshire, UK, 2001.
  - 58.Forgas, J.P. Mood effects on decision making strategies. *Aust. J. Psychol.*, 1989, 41: 197–214.
  59. Loewenstein, G.; Lerner, J.S. The role of affect in decision making. In *Handbook of Affective Science*; Davidson, R.; Goldsmith, H.; Scherer, K., Eds.; Oxford University Press: Oxford, 2003; pp. 619–642.
  60. Gliozzo, A.; Ackerson, C.; Bhattacharya, R.; et al. *Building Cognitive Applications with IBM Watson Services: Volume 1 Getting Started*; IBM Redbooks, IBM Garage, United States, 2017. Available online:<https://www.redbooks.ibm.com/abstracts/sq248387.html> (accessed on 6 November 2022).
  61. Iyengar, S.S.; Mukhopadhyay, S.; Steinmuller, C.; et al, Preventing future oil spills with software-based event detection. *Computer*, 2010, 43: 95–97.
  62. Commissiong, M.A. Student Engagement, Self-Regulation, Satisfaction, and Success in Online Learning Environments. Ph.D. Thesis, Walden University, Minneapolis, USA, 2020.
  63. Starkey, K.; Hatchuel, A.; Tempest, S, Management research and the new logics of discovery and engagement. *J. Manage. Stud.*, 2009, 46: 547–558.
  64. Araujo, T.; Helberger, N.; Kruikemeier, S.; et al, In AI we trust? Perceptions about automated decision-making by artificial intelligence *AI Soc.*, 2020, 35: 611–623.
  65. Chirapurath, J. *Knowledge Mining: The Next Wave of Artificial Intelligence-Led Transformation*; Harvard Business Review, Harvard Business Publishing, Watertown, Massachusetts, 21.11.2019. Available online:<https://hbr.org/sponsored/2019/11/knowledge-mining-the-next-wave-of-artificial-intelligence-led-transformation> (accessed on 11 November 2022).
  66. Sagiroglu, S.; Sinanc, D. Big data: A review. In *Proceedings of 2013 International Conference on Collaboration Technologies and Systems (CTS), San Diego, CA, USA, 20–24 May 2013*; IEEE: San Diego, USA, 2013; pp. 42–47. doi:10.1109/CTS.2013.6567202
  67. Sarker, I.H, Machine learning: Algorithms, real-world applications and research directions. *SN Comput. Sci.*, 2021, 2: 160.
  68. Peng, W.; Varanka, T.; Mostafa, A.; et al, Hyperbolic deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022, 44: 10023–10044.
  69. Li, Z.M.; Tian, W.W.; Li, Y.T.; et al. A more effective method for image representation: Topic model based on latent dirichlet allocation. In *Proceedings of 2015 14th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics), Xi'an, China, 26–28 August 2015*; IEEE: Xi'an, China, 2015; pp. 143–148. doi:10.1109/CADGRAPHICS.2015.19
  70. Jarrahi, M.H, Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making. *Bus. Horiz.*, 2018, 61: 577–586.
  71. Wyant, D.K.; Bingi, P.; Knight, J.R.; et al. DeTER framework: A novel paradigm for addressing cybersecurity concerns in mobile healthcare. In *Research Anthology on Securing Medical Systems and Records*; Information Resources Management Association, Ed.; IGI Global, Chocolate Ave. Hershey, PA 17033, USA, 2022; pp. 381–407. doi:10.4018/978-1-6684-6311-6.ch019
  72. Arrieta, A.B.; Diaz-Rodriguez, N.; Del Ser, J.; et al, Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion*, 2020, 58: 82–115.
  73. Holzinger, A, Explainable AI and multi-modal causability in medicine. *i-com*, 2021, 19: 171–179.
  74. Ravindran, N.J.; Gopalakrishnan, P. Predictive analysis for healthcare sector using big data technology. In *Proceedings of 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), Bangalore, India, 16–18 August 2018*; IEEE: Bangalore, India, 2018; pp. 326–331. doi:10.1109/ICGCIoT.2018.8753090
  75. Lin, S.J.; Hsu, M.F. Incorporated risk metrics and hybrid AI techniques for risk management. *Neural Comput. Appl.* 2017, 28, 3477–3489. doi:10.1007/s00521-016-2253-4
  76. Peeters, M.M.M.; van Diggelen, J.; van den Bosch, K.; et al, Hybrid collective intelligence in a human–AI society. *AI Soc.*, 2021, 36: 217–238.
  77. Jobin, A.; Ienca, M.; Vayena, E, The global landscape of AI ethics guidelines. *Nat. Mach. Intell.*, 2019, 1: 389–399.

**Citation:** Zhao, G.; Li, Y.; Xu, Q. From Emotion AI to Cognitive AI *International Journal of Network Dynamics and Intelligence*. <https://doi.org/10.53941/ijndi0101006>

**Publisher’s Note:** Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license <https://creativecommons.org/licenses/by/4.0/>.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/367284460>

# Development of a Basic Chemistry Conversational Corpus

Article · January 2023

DOI: 10.31080/ASNH.2023.07.1187

---

CITATIONS

0

READS

15

3 authors, including:



Maurice HT Ling

Temasek Polytechnic

109 PUBLICATIONS 587 CITATIONS

[SEE PROFILE](#)



Poh Nguk Lau

Temasek Polytechnic

7 PUBLICATIONS 15 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Science and Education Portraits [View project](#)



Computational and Statistical Tools for Research [View project](#)



## Development of a Basic Chemistry Conversational Corpus

Maurice HT Ling<sup>1\*</sup>, Syameer Mustakim<sup>1</sup> and Poh Nguk Lau<sup>1,2</sup>

<sup>1</sup>School of Applied Sciences, Temasek Polytechnic, Singapore

<sup>2</sup>Learning Academy, Temasek Polytechnic, Singapore

\*Corresponding Author: Maurice HT Ling, School of Applied Sciences, Temasek Polytechnic, Singapore.

Received: December 31, 2022

Published: January 13, 2023

© All rights are reserved by Maurice HT Ling, et al.

### Abstract

Chatbot technology can be an important tool and supplement to education, leading to explorations in this area. Corpus-based chatbot building has a relatively low entry barrier as it only requires a relevant corpus to train a chatbot engine. The corpus is a set of human-readable questions and answers and may be an amalgamation of existing corpora. However, a suitable chemistry-based chatbot corpus catering for a freshman general chemistry course addressing inorganic and physical chemistry has not been developed. In this study, we present a basic chemistry conversational corpus consisting of 998 pairs of questions and answers, focused on a freshman general chemistry course addressing inorganic and physical chemistry. Ten human raters evaluated the responses of a chatbot trained on the corpus and suggests that the corpus resulted in better response than random ( $t = 17.4$ ,  $p$ -value = 1.86E-53). However, only 20 of the 50 test questions show better responses compared to random (difference in mean score  $\geq 1.9$ , paired t-test  $p$ -value  $\leq 0.0324$ ), suggesting that the corpus provides better responses to certain questions rather than overall better responses, with questions related to definitions and computational procedures answered more accurately. Hence, this provides a baseline for future corpora development.

**Keywords:** Chemistry; Conversational Corpus; ChatterBot

### Introduction

Chatbot can be defined as a computer program that mimics human conversation [1] and has its roots in Turing Test [2]. In recent years, chatbots have been used in a variety of fields [3]; such as commerce [4], healthcare [5], and education [6-9]; especially during COVID-19 pandemic as a means to deliver services [10-12]. Fonna and Widayantoro [13] found that chatbots incorporated into tutorials can supplement pharmacology education. Kovacek and Chow [14] presented a chatbot catering to radiation safety education. Several studies also pointed to favourable student learning outcomes when chatbot are deployed to teaching and learning. For example, Atmosukarto., et al. [15] showed that by incorporating chatbot to an online chemistry course, course completion rate can be improved as the chatbot can cater to student's on-demand query needs, as compared to the limited online presence of a human tutor. Chatbots could also be used as a preparatory resource to complement online courses to augment learners' readiness for high-stakes assessment [16]. The interactive affordance and immediate feedback features provided by chatbots also facilitate self-directed learning and self-evaluation [17].

Of the 6 paradigms of chatbot building classified by Luo., et al. [3]; namely, template-based, corpus-based, intent-based, recurrent neural network-based, reinforcement learning-based, and hybrid; corpus-based appears to be the easiest and most flexible. A chat corpus can be built from existing questions and answers pair and corpus-based paradigm will allow a new chatbot to be implemented by retraining it on a new conversational corpus [18,19], which had been successfully demonstrated [20-22]. Another important advantage is that a corpus can be incrementally built or amalgamation of existing corpora. Hence, a chatbot can be considered as a chatbot engine (which can be defined as a software component that accepts a natural human language input, processes, and respond with an output in a natural human language [23]) trained on a corpus. Therefore, the advancement of chatbot technology and use cases can be the parallel track of advancement of chatbot engine technology, and the advancement of corpus.

In this study, we present a basic chemistry conversational corpus of 998 pairs of questions and answers, focused on a tertiary-level freshman inorganic and physical chemistry course; as such a corpus has not been widely developed. Ten human raters were

used to evaluate the responses of a chatbot trained on the corpus and suggests that the corpus resulted in better response than random ( $t = 17.4$ , p-value = 1.86E-53). Hence, this provides a baseline for future corpora development.

## Materials and Methods

### Corpus development

A conversational chemistry corpus, with contents based on a freshman general chemistry course addressing inorganic and physical chemistry was built from existing frequently asked questions (FAQs). Senior year students and faculty members were asked to contribute questions and answers related to the course. These questions and answers were converted into YAML format (<https://yaml.org/>).

### Corpus testing

The corpus was used to train a chatbot, which was built on Chatterbot (<https://chatterbot.readthedocs.io>) [23] and known as ChemBot. An untrained chatbot, known as DumbBot, was used for comparison. A set of 50 questions were posed to both ChemBot and DumbBot and each of the replies were evaluated by 10 human raters for quality [24]. The 50 questions were sourced from the 10 human raters to represent questions that may be asked by freshmen. The overall mean scores, mean scores by raters, and mean scores by questions were evaluated as mean scores is the most common metric of academic performance [25,26].

### Statistical analysis

T-test assuming unequal variance was used to compare the mean scores between ChemBot and DumbBot. 1-way ANOVA was used to compare mean scores between the raters and between questions. Paired t-test was used to compare the mean scores, which were paired by question or by rater. A p-value of less than 0.05 was statistically significant.

## Results and Discussion

### Corpus improves chatbot responses

A total of 998 pairs of questions and answers were compiled as corpus, which was then evaluated by comparing a trained chatbot (ChemBot) against an untrained chatbot (DumbBot). Each of the 10 raters scored the replies from both ChemBot and DumbBot on 50 questions. Mean scores were obtained by aggregating scores of the 50 questions per rater across the 10 raters (giving a total of 500 scores). Table 1 presents the mean score and standard error rated on ChemBot and DumbBot.

The mean score of ChemBot is 3.75 with standard error of 0.158 while the mean score of DumbBot is 1.01 with standard error of 0.004, the difference of which is statistically significant using t-

	ChemBot	DumbBot
Mean score	3.75	1.01
Standard error	0.158	0.004

**Table 1:** Descriptive statistics for answer accuracy for ChemBot and DumbBot.

test assuming unequal variances ( $t = 17.4$ , p-value = 1.86E-53). This suggests that the developed corpus improves the response of ChemBot when compared to DumbBot, which is supported by previous studies [19,21,27-29] and is the fundamental motivation for corpus development [20,30].

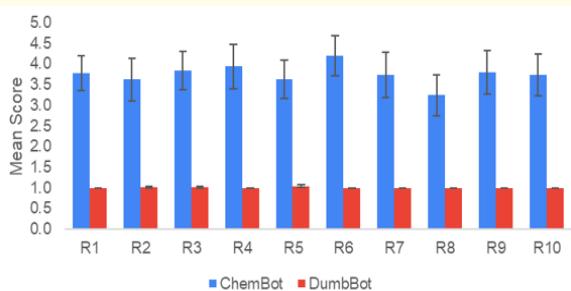
### Comparable scores by each ratter on chembot

Of the 10 raters, rater R8 is the only chemistry lecturer (last author) with the rest being students. The mean score of ChemBot ( $n = 50$ ) by each rater ranged from 3.24 with standard error of 0.504 (by rater R8) to 4.20 with standard error of 0.481 (by rater R6). Although the scoring between rater R6 and R8 is significant on paired t-test as paired by questions (p-value = 9.73E-3); the correlation of scores given by rater R6 and R8 is 0.738, which is higher than that between rater R1 and R8 (see Figure 1). Interestingly, paired t-test suggests that the scores given by rater R1 and R8 is not significant (p-value = 0.144) despite having the lowest correlation. 2-samples t-test assuming unequal variances between rater R1 and R8 (p-value = 0.416) and between rater R6 and R8 (p-value 0.171) are not significant. This is supported by 1-way ANOVA suggesting no significance in the mean scores across all raters (Figure 2;  $F = 0.238$ , p-value = 0.989). For DumbBot, 1-way ANOVA also suggests no significance in the mean scores across all raters (Figure 2;  $F = 1.235$ , p-value = 0.271).

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
R1	1.000	0.843	0.935	0.824	0.800	0.774	0.791	0.706	0.804	0.838
R2	0.843	1.000	0.932	0.898	0.890	0.851	0.936	0.880	0.932	0.949
R3	0.935	0.932	1.000	0.863	0.849	0.822	0.864	0.808	0.868	0.894
R4	0.824	0.898	0.863	1.000	0.861	0.881	0.882	0.834	0.924	0.945
R5	0.800	0.890	0.849	0.861	1.000	0.889	0.885	0.750	0.838	0.866
R6	0.774	0.851	0.822	0.881	0.889	1.000	0.830	0.738	0.815	0.853
R7	0.791	0.936	0.864	0.882	0.885	0.830	1.000	0.796	0.866	0.881
R8	0.706	0.880	0.808	0.834	0.750	0.738	0.796	1.000	0.900	0.901
R9	0.804	0.932	0.868	0.924	0.838	0.815	0.866	0.900	1.000	0.970
R10	0.838	0.949	0.894	0.945	0.866	0.853	0.881	0.901	0.970	1.000

**Figure 1:** Correlation matrix between raters.

The minimum Pearson's correlation coefficient is 0.706 ( $n = 50$ ,  $t = 6.907$ , p-value = 1.021E-8) between R8 and R1. The minimum correlation among student raters (R8 is a lecturer) is 0.774 ( $n = 50$ ,  $t = 8.469$ , p-value = 4.346E-11).



**Figure 2:** Mean scores by each ratter  
The error bars denote standard errors.

After removing rater R8, the mean score of ChemBot ( $n = 50$ ) by each student rater ranged from 3.62 with standard error of 0.523 (by rater R2) to 4.20 with standard error of 0.481 (by rater R6).

2-samples t-test assuming unequal variances between rater R2 and R6 is not significant ( $p\text{-value} = 0.416$ ) with 1-way ANOVA suggesting no significance in the mean scores across all 9 student raters (Figure 2;  $F = 0.122$ ,  $p\text{-value} = 0.998$ ). For DumbBot, a 1-way ANOVA also suggests no significance in the mean scores across all raters (Figure 2;  $F = 1.200$ ,  $p\text{-value} = 0.297$ ). Taken together, the scores across all 10 raters are comparable.

### Mean scores differ by questions

The mean score of ChemBot ( $n = 10$ ) by question ranged from 1.00 with standard error of zero for 13 questions (Questions 7, 11, 18, 23, 24, 25, 27, 32, 33, 43, 44, 45, and 50) to 9.10 with standard error of 0.233 for question 26 (see Table 2 and Figures 3A to 3C). This is supported by 1-way ANOVA suggesting significant differences in the mean scores across questions (Figure 3;  $F = 59.939$ ,  $p\text{-value} = 1.1E-166$ ).

Question	ChemBot Mean Score	ChemBot Median	DumbBot Mean Score
Q1. What is the electronic configuration of nitrogen? *	5.50 (0.992)	7.5	1.00 (0.000)
Q2. What are the forces between ammonia?	1.70 (0.616)	1.0	1.00 (0.000)
Q3. Why do Protons only exist in the nucleus? *	3.60 (0.653)	5.0	1.00 (0.000)
Q4. Why are ionic bonds the strongest bond?	2.10 (0.737)	1.0	1.00 (0.000)
Q5. What are orbitals? *	3.90 (0.936)	3.5	1.00 (0.000)
Q6. How to calculate pH of a solution?	1.40 (0.267)	1.0	1.00 (0.000)
Q7. What are the bonds present in a covalent bonding?	1.00 (0.000)	1.0	1.00 (0.000)
Q8. What is hydrogen bonding?	2.00 (0.516)	1.0	1.00 (0.000)
Q9. Why do ionic compounds have a high boiling point? *	8.50 (0.307)	8.0	1.00 (0.000)
Q10. Why are some acids strong and some are weak acid?	1.30 (0.213)	1.0	1.00 (0.000)
Q11. Why water have hydrogen bonding?	1.00 (0.000)	1.0	1.00 (0.000)
Q12. What is the classical method of naming compounds?	1.60 (0.499)	1.0	1.00 (0.000)
Q13. What is the equilibrium constant? *	8.60 (0.600)	9.0	1.00 (0.000)
Q14. What is ionic equilibrium?	2.20 (0.727)	1.0	1.00 (0.000)
Q15. How to calculate molarity? *	8.70 (0.335)	8.5	1.00 (0.000)
Q16. What are the different types of bonding forces?	1.50 (0.342)	1.0	1.00 (0.000)
Q17. What is a buffer solution?	1.20 (0.133)	1.0	1.00 (0.000)
Q18. How do I make a buffer solution?	1.00 (0.000)	1.0	1.00 (0.000)
Q19. What is equilibrium? *	8.40 (0.221)	8.5	1.00 (0.000)
Q20. How to find dilution factor? *	7.60 (0.542)	8.0	1.00 (0.000)
Q21. How to calculate mol?	1.20 (0.133)	1.0	1.00 (0.000)
Q22. What is atomic mass?	2.90 (0.752)	2.0	1.00 (0.000)
Q23. What is atomic weight?	1.00 (0.000)	1.0	1.00 (0.000)
Q24. What is molecular weight?	1.00 (0.000)	1.0	1.00 (0.000)
Q25. What is formula weight?	1.00 (0.000)	1.0	1.00 (0.000)
Q26. How to calculate percentage composition? *	9.10 (0.233)	9.0	1.00 (0.000)
Q27. What is the Avogadro's number?	1.00 (0.000)	1.0	1.00 (0.000)
Q28. What is molar mass? *	8.50 (0.543)	9.0	1.00 (0.000)
Q29. How do I find the number of moles? *	8.30 (0.633)	9.0	1.00 (0.000)

Q30. What is stoichiometry? *	8.00 (0.558)	8.5	1.00 (0.000)
Q31. What is a limiting reactant? *	8.70 (0.335)	9.0	1.00 (0.000)
Q32. What is an excess reactant?	1.00 (0.000)	1.0	1.00 (0.000)
Q33. How to find precent yield?	1.00 (0.000)	1.0	1.00 (0.000)
Q34. What is a monoprotic acid?	1.20 (0.133)	1.0	1.00 (0.000)
Q35. What is amphoteric?	1.40 (0.306)	1.0	1.00 (0.000)
Q36. What is the unit of pH?	1.50 (0.401)	1.0	1.00 (0.000)
Q37. What do pH and pOH measure?	1.10 (0.100)	1.0	1.00 (0.000)
Q38. What happens when a strong acid and a base react?	1.20 (0.200)	1.0	1.00 (0.000)
Q39. What is a conjugate base and conjugate acid?	1.20 (0.200)	1.0	1.20 (0.133)
Q40. What is buffer capacity? *	8.60 (0.306)	8.5	1.00 (0.000)
Q41. What is equivalence point? *	8.50 (0.269)	8.5	1.00 (0.000)
Q42. What is chemical kinetics? *	8.50 (0.500)	9.0	1.00 (0.000)
Q43. What is activation energy?	1.00 (0.000)	1.0	1.20 (0.133)
Q44. What does a catalyst do?	1.00 (0.000)	1.0	1.00 (0.000)
Q45. How does the temperature affect the reaction rate?	1.00 (0.000)	1.0	1.00 (0.000)
Q46. What is a reversible reaction? *	7.90 (0.458)	8.0	1.00 (0.000)
Q47. What is a dynamic equilibrium? *	7.30 (0.943)	8.0	1.00 (0.000)
Q48. What is Le Chatelier's principle? *	8.70 (0.300)	9.0	1.00 (0.000)
Q49. Why does the equilibrium shifts?	1.10 (0.100)	1.0	1.00 (0.000)
Q50. What is a weak acid?	1.00 (0.000)	1.0	1.00 (0.000)

**Table 2:** Mean score by question in ChemBot and DumbBot. Standard error in brackets.

\*mean scores significantly higher in ChemBot than Dumbot (paired t-test difference in mean score  $\geq 1.9$ , p-value  $\leq 0.0324$ ).

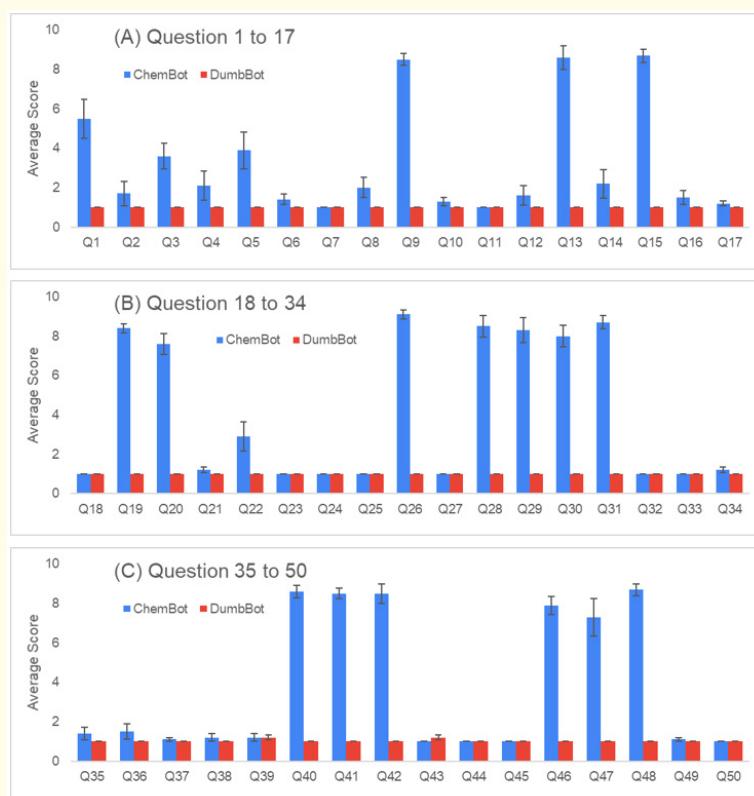
For DumbBot, only 2 questions (Questions 39, and 43) have a mean score of more than zero (mean score = 1.20 with standard error of 0.133, see Figure 3C). Interestingly, the mean score for question 43 is higher in DumbBot (mean score = 1.20) than ChemBot (mean score = 1.00); however, this difference is not significant (paired t-test p-value = 0.168).

Of the 50 questions, the mean scores of 20 questions (Questions 1, 3, 5, 9, 13, 15, 19, 20, 22, 26, 28, 29, 30, 31, 40, 41, 42, 46, 47, and 48) were significantly higher in ChemBot as compared to DumbBot (difference in mean score  $\geq 1.9$ , paired t-test p-value  $\leq 0.0324$ ). This suggests that ChemBot provides better responses to certain questions rather than overall better responses, which has been previously demonstrated [31]. This is plausible as Callejas-Rodríguez, *et al.* [32] had shown that chatbot personality may be generated from training corpus. In general, it is also noted that ChemBot handles closed-ended concepts such as direct definitions (Q40, 41 and 42) and those related to computational procedures (such as Q15, 26, 29) more accurately. Questions that are more diffused, demand more in-depth explanation (Q18, Q45), has “compare/contrast” requirements (Q10 and Q39) or more than one keyword (such as “electronic configuration” and “nitrogen” in Q1; “atomic” and

“mass” in Q22 and Q23) are typically poorly answered. Previous studies have demonstrated the impact of keyword recognition in the accuracy of question and answer generated by chatbot applications [17].

### Applications and future work

This baseline corpus could be integrated into a chatbot application currently on trial within the institution. This application is a commercial solution purchased on a licence-basis and requires a direct feed of FAQs for training purposes. The chatbot is integrated with an institution-wide communications interface, the Microsoft Teams, on which learners could pose a question to the chatbot. Instead of directly providing an answer, the chatbot engine directs another learner in the class to provide the answer. The instructor could choose to rate the learner-provided answer, archive a good answer into the FAQ database for future use or provide a more accurate response from the original database. Two clear advantages are apparent. One, the learner actively contributes to the learning of their peers, and secondly, the corpus could also be enhanced. Such a chatbot could then be implemented and tested in a chemistry course.



**Figure 3:** Mean scores across questions.

The error bars denote standard errors. Panel A shows mean scores from questions 1 to 17. Panel B shows mean scores from questions 18 to 34. Panel C shows mean scores from questions 35 to 50.

## Conclusion

Here, we present a basic chemistry conversational corpus consisting of 998 pairs of questions and answers, focused on tertiary year one inorganic and physical chemistry course; which was used to train chatbot based on Chatterbot engine and shown to generate better chatbot responses than untrained chatbot ( $t = 17.4$ ,  $p\text{-value} = 1.86E-53$ ). However, only 20 of the 50 test questions show better responses compared to random (paired t-test  $p\text{-value} \leq 0.0324$ ), suggesting that the corpus training results in better responses to certain questions rather than overall better responses. Hence, this study provides a baseline for future corpus development.

## Supplementary Materials

Materials from this study can be downloaded at [https://bit.ly/ChemBot\\_1](https://bit.ly/ChemBot_1). Video showing comparative testing of ChemBot\_1 and DumbBot can be found at <https://youtu.be/tEJVRFphtLE>.

## Acknowledgement

This work is supported by Temasek Polytechnic School of Applied Science under the Student Project Fund (TP\_PR1199).

## Conflict of Interest

The authors declare no conflict of interest.

## Bibliography

1. Adamopoulou E and Moussiades L. "An Overview of Chatbot Technology". AIAI 2020: Artificial Intelligence Applications and Innovations, eds Maglogiannis I, Iliadis L, Pimenidis E (Springer International Publishing, Cham) (2020): 373-383.
  2. Turing AM. "Computing Machinery and Intelligence". *Mind* LIX 236 (1950): 433-460.
  3. Luo B, et al. "A Critical Review of State-of-the-Art Chatbot Designs and Applications". *WIREs Data Mining and Knowledge Discovery* 12 (2022): e1434.
  4. Landim ARDB. "Chatbot Design Approaches for Fashion E-commerce: An Interdisciplinary Review". *International Journal of Fashion Design, Technology and Education* 15.2 (2022): 200-210.
  5. Tjiptomongsoguno ARW. "Medical Chatbot Techniques: A Review". Software Engineering Perspectives in Intelligent Systems, Advances in Intelligent Systems and Computing, eds Silhavy R, Silhavy P and Prokopova Z. "(Springer International Publishing, Cham) 1294 (2020): 346-356.

6. Okonkwo CW and Ade-Ibijola A. "Chatbots Applications in Education: A Systematic Review". *Computers and Education: Artificial Intelligence* 2 (2021): 100033.
7. Yang S and Evans C. "Opportunities and Challenges in Using AI Chatbots in Higher Education". Proceedings of the 2019 3rd International Conference on Education and E-Learning (ACM, Barcelona Spain) (2019): 79-83.
8. Hamam D. "The New Teacher Assistant: A Review of Chatbots' Use in Higher Education". HCI International 2021 - Posters, Communications in Computer and Information Science., eds Stephanidis C, Antona M, Ntoa S (Springer International Publishing, Cham) 1421 (2021): 59-63.
9. Akinwalere SN and Ivanov V. "Artificial Intelligence in Higher Education: Challenges and Opportunities". *Border Crossing* 12.1 (2022): 1-15.
10. Amiri P and Karahanna E. "Chatbot Use Cases in the Covid-19 Public Health Response. *Journal of the American Medical Informatics Association* 29.5 (2022): 1000-1010.
11. Zhu Y, et al. "It Is Me, Chatbot: Working to Address the COVID-19 Outbreak-Related Mental Health Issues in China. User Experience, Satisfaction, and Influencing Factors". *International Journal of Human-Computer Interaction* 38.12 (2022): 1182-1194.
12. Sweidan SZ., et al. "SIAAA-C: A Student Interactive Assistant Android Application with Chatbot During COVID-19 Pandemic". *Computer Applications in Engineering Education* 29.6 (2021): 1718-1742.
13. Fonna MR and Widayantoro DH. "Tutorial System in Learning Activities Through Machine Learning-Based Chatbot Applications in Pharmacology Education". 2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA) (IEEE, Bandung, Indonesia) (2019): 1-6.
14. Kovacek D and Chow JCL. "An AI-Assisted Chatbot for Radiation Safety Education in Radiotherapy". *IOP SciNotes* 2.3 (2021): 034002.
15. Atmosukarto I., et al. "Enhancing Adaptive Online Chemistry Course with AI-Chatbot". 2021 IEEE International Conference on Engineering, Technology and Education (TALE) (2021): 838-843.
16. Korsakova E., et al. "Chemist Bot as a Helpful Personal Online Training Tool for the Final Chemistry Examination". *Journal of Chemical Education* 99.2 (2022): 1110-1117.
17. Mahroof A., et al. "An AI based Chatbot to Self-Learn and Self-Assess Performance in Ordinary Level Chemistry. 2020 2nd International Conference on Advancements in Computing (ICAC) (IEEE, Malabe, Sri Lanka) (2020): 216-221.
18. Shawar BAA. "A Corpus Based Approach to Generalise a Chatbot System". Doctor of Philosophy (University of Leeds, School of Computing) (2005).
19. Shawar BA and Atwell ES. "Using Corpora in Machine-Learning Chatbot Systems". *International Journal of Corpus Linguistics* 10.4 (2005): 489-516.
20. Rikters M., et al. "Designing the Business Conversation Corpus". Proceedings of the 6th Workshop on Asian Translation (Association for Computational Linguistics, Hong Kong, China), 54-61.
21. Shawar BA and Atwell E. "Using the Corpus of Spoken Afrikaans to Generate an Afrikaans Chatbot". *Southern African Linguistics and Applied Language Studies* 21.4 (2003): 283-294.
22. Shawar BA and Atwell E. "Arabic Question-Answering via Instance Based Learning from an FAQ Corpus". Proceedings of the CL 2009 International Conference on Corpus Linguistics". UCREL 386 (2016): 1-12.
23. Sim KS and Ling MH. "Installation and Documentation Evaluation of Recent (01 January 2020 to 15 February 2021) Chatbot Engines from Python Package Index (PyPI)". *Acta Scientific Computer Sciences* 3.8 (2011): 38-43.
24. Shawar BA and Atwell E. "Different measurements metrics to evaluate a chatbot system". (Association for Computational Linguistics) (2007): 89-96.
25. Kumar S., et al. "Defining and Measuring Academic Performance of Hei Students - A Critical Review". *Turkish Journal of Computer and Mathematics Education* 12.6 (2021): 3091-3105.
26. Alyahyan E and Düşteğör D. "Predicting Academic Success in Higher Education: Literature Review and Best Practices". *International Journal of Educational Technology in Higher Education* 17.1 (2020): 3.

27. Kim J., et al. "Two-Step Training and Mixed Encoding-Decoding for Implementing a Generative Chatbot with a Small Dialogue Corpus. Proceedings of the Workshop on Intelligent Interactive Systems and Language Generation (2IS and NLG) (Association for Computational Linguistics, Tilburg, the Netherlands) (2018): 31-35.
28. Kowsher Md., et al. "Doly: Bengali Chatbot for Bengali Education". 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (IEEE, Dhaka, Bangladesh) (2019): 19.
29. Blanc C., et al. "FlauBERT vs. CamemBERT: Understanding Patient's Answers by a French Medical Chatbot". *Artificial Intelligence in Medicine* 127 (2022): 102264.
30. Shawar BAA and Atwell E. "Automatic Extraction of Chatbot Training Data from Natural Dialogue Corpora" (2016): 29-38.
31. Kapočiūtė-Dzikienė J. "A Domain-Specific Generative Chatbot Trained from Little Data". *Applied Sciences* 10.7 (2020): 2221.
32. Callejas-Rodríguez Á., et al. "From Dialogue Corpora to Dialogue Systems: Generating a Chatbot with Teenager Personality for Preventing Cyber-Pedophilia. Text, Speech, and Dialogue, Lecture Notes in Computer Science., eds Sojka P, Horák A, Kopeček I, Pala K (Springer International Publishing, Cham) 9924 (2016): 531-539.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/335395532>

# Deep Learning Based Chatbot Models

Preprint · November 2017

---

CITATIONS

0

READS

956

1 author:



Richard Csaky

University of Oxford

10 PUBLICATIONS 60 CITATIONS

SEE PROFILE

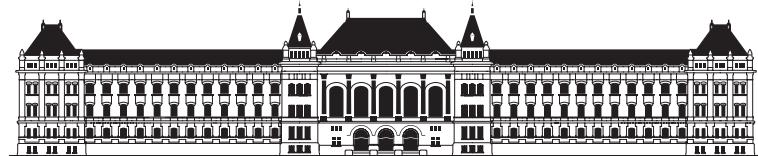
Some of the authors of this publication are also working on these related projects:



Protein based logic circuits [View project](#)



Dialog agents [View project](#)



BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS  
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS  
DEPARTMENT OF AUTOMATION AND APPLIED INFORMATICS

# Deep Learning Based Chatbot Models

## SCIENTIFIC STUDENTS' ASSOCIATIONS REPORT

Author:  
Richárd Krisztián Csáky

Supervised by  
Gábor Recski

2017

## Abstract

A conversational agent (chatbot) is a piece of software that is able to communicate with humans using natural language. Modeling conversation is an important task in natural language processing and artificial intelligence (AI). Indeed, ever since the birth of AI, creating a good chatbot remains one of the field's hardest challenges. While chatbots can be used for various tasks, in general they have to understand users' utterances and provide responses that are relevant to the problem at hand.

In the past, methods for constructing chatbot architectures have relied on hand-written rules and templates or simple statistical methods. With the rise of deep learning these models were quickly replaced by end-to-end trainable neural networks around 2015. More specifically, the recurrent encoder-decoder model [Cho et al., 2014] dominates the task of conversational modeling. This architecture was adapted from the neural machine translation domain, where it performs extremely well. Since then a multitude of variations [Serban et al., 2016] and features were presented that augment the quality of the conversation that chatbots are capable of.

In my work, I conduct an in-depth survey of recent literature, examining over 70 publications related to chatbots published in the last 3 years. Then I proceed to make the argument that the very nature of the general conversation domain demands approaches that are different from current state-of-the-art architectures. Based on several examples from the literature I show why current chatbot models fail to take into account enough priors when generating responses and how this affects the quality of the conversation. In the case of chatbots these priors can be outside sources of information that the conversation is conditioned on like the persona [Li et al., 2016a] or mood of the conversers. In addition to presenting the reasons behind this problem, I propose several ideas on how it could be remedied.

The next section of my paper focuses on adapting the very recent Tranformer [Vaswani et al., 2017] model to the chatbot domain, which is currently the state-of-the-art in neural machine translation. I first present my experiments with the vanilla model, using conversations extracted from the Cornell Movie-Dialog Corpus [Danescu-Niculescu-Mizil and Lee, 2011]. Secondly, I augment the model with some of my ideas regarding the issues of encoder-decoder architectures. More specifically, I feed additional features into the model like mood or persona together with the raw conversation data. Finally, I conduct a detailed analysis of how the vanilla model performs on conversational data by comparing it to previous chatbot models and how the additional features, affect the quality of the generated responses.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>History of Chatbots</b>	<b>5</b>
2.1	Modeling Conversations . . . . .	5
2.2	Early Approaches . . . . .	6
2.3	The Encoder-Decoder Model . . . . .	7
2.3.1	Recurrent Neural Networks . . . . .	8
2.3.2	The Seq2seq Model . . . . .	9
2.3.3	Deep Seq2seq Models . . . . .	11
2.3.4	Decoding and Vocabulary . . . . .	11
<b>3</b>	<b>Background</b>	<b>13</b>
3.1	Further Details of Encoder-Decoder Models . . . . .	13
3.1.1	Context . . . . .	13
3.1.2	Objective Functions . . . . .	14
3.1.3	Evaluation Methods . . . . .	16
3.2	Augmentations To The Encoder-Decoder Model . . . . .	17
3.2.1	Attention . . . . .	17
3.2.2	Pretraining . . . . .	20
3.2.3	Additional Input Features . . . . .	21
3.2.4	Knowledge Bases and Copying . . . . .	23
3.3	Different Approaches to Conversational Modeling . . . . .	25
3.3.1	Hierarchical Models . . . . .	25
3.3.2	Task-Oriented Dialog Systems . . . . .	27
3.3.3	Reinforcement Learning . . . . .	29
3.3.4	Different Encoder-Decoder Models . . . . .	32
3.4	Criticism . . . . .	33
3.4.1	Datasets . . . . .	33
3.4.2	The Loss Function . . . . .	34
3.4.3	Memory . . . . .	34
3.4.4	Evaluation Metrics . . . . .	35
3.5	Summary . . . . .	36
<b>4</b>	<b>Experiments</b>	<b>37</b>
4.1	The Transformer Model . . . . .	37
4.1.1	Encoder and Decoder Networks . . . . .	38
4.1.2	Attention Mechanisms . . . . .	38
4.1.3	Feed-Forward Networks . . . . .	39
4.1.4	Positional Encoding . . . . .	40
4.1.5	Regularization and Other Techniques . . . . .	40

4.2	Datasets . . . . .	40
4.2.1	Cornell Movie-Dialog Corpus . . . . .	41
4.2.2	OpenSubtitles Corpus . . . . .	41
4.3	Training Details . . . . .	41
4.3.1	Tensor2Tensor . . . . .	42
4.3.2	Cornell Movie Training . . . . .	42
4.3.3	Cornell Movie Training with Speakers . . . . .	42
4.3.4	OpenSubtitles Training . . . . .	43
4.3.5	OpenSubtitles Training Finetuned with Cornell Movie Data . . . . .	43
<b>5</b>	<b>Results</b>	<b>44</b>
5.1	Quantitative Analysis . . . . .	44
5.2	Qualitative Analysis . . . . .	46
<b>6</b>	<b>Future Work</b>	<b>52</b>
6.1	Ideas Towards Solving The Loss Function Issue . . . . .	52
6.2	Temporal Conditioning and Memory . . . . .	54
6.3	Additional Ideas . . . . .	55
<b>7</b>	<b>Conclusion</b>	<b>56</b>
	<b>References</b>	<b>57</b>

# 1 Introduction

A conversational agent (chatbot) is a piece of software that is able to communicate with humans using natural language. Ever since the birth of AI, modeling conversations remains one of the field’s toughest challenges. Even though they are far from perfect, chatbots are now used in a plethora of applications like Apple’s Siri [Apple, 2017], Google’s Google Assistant [Google, 2017] or Microsoft’s Cortana [Microsoft, 2017a]. In order to fully understand the capabilities and limitations of current chatbot architectures and techniques an in-depth survey is conducted, where related literature published over the past 3 years is examined and a recently introduced neural network model is trained using conversational data.

The paper begins with a brief overview of the history of chatbots in Section 2, where the properties and objectives of conversational modeling are discussed. Early approaches are presented in Section 2.2 as well as the current dominating model for building conversational agents, based on neural networks, in Section 2.3.

In Section 3 key architectures and techniques are described that were developed over the past 3 years related to chatbots. Publications are grouped into categories on the basis of specific techniques or approaches discussed by the authors. After this, criticism is presented in Section 3.4 regarding some of the properties of current chatbot models and it is shown how several of the techniques used are inappropriate for the task of modeling conversations.

In the next part (Section 4) preliminary experiments are conducted by training a novel neural network based model, the Transformer [Vaswani et al., 2017], using dialog datasets [Danescu-Niculescu-Mizil and Lee, 2011, Tiedemann, 2009, Lison and Tiedemann, 2016]. Several trainings are run using these datasets, detailed in Section 4.3. The results of the various training setups are presented in Section 5 by qualitatively comparing them to previous chatbot models and by using standard evaluation metrics.

Finally, in Section 6 possible directions for future work are offered. More specifically, several ideas are proposed in order to remedy the problems presented in Section 3.4 and future research directions are discussed.

## 2 History of Chatbots

### 2.1 Modeling Conversations

Chatbot models usually take as input natural language sentences uttered by a user and output a response. There are two main approaches for generating responses. The traditional approach is to use hard-coded templates and rules to create chatbots, presented in Section 2.2. The more novel approach, discussed in detail in Section 2.3, was made possible by the rise of deep learning. Neural network models are trained on large amounts of data to learn the process of generating relevant and grammatically correct responses to input utterances. Models have also been developed to accommodate for spoken or visual inputs. They oftentimes make use of a speech recognition component to transform speech into text [Serban et al., 2017b] or convolutional neural networks [Krizhevsky et al., 2012] that transform the input pictures into useful representations for the chatbot [Havrylov and Titov, 2017]. The latter models are also called visual dialog agents, where the conversation is grounded on both textual and visual input [Das et al., 2017].

Conversational agents exist in two main forms. The first one is the more traditional task-oriented dialog system, which is limited in its conversational capabilities, however it is very robust at executing task specific commands and requirements. Task-oriented models are built to accomplish specific tasks like making restaurant reservations [Joshi et al., 2017, Bordes et al., 2016] or promoting movies [Yu et al., 2017], to name a few. These systems often don't have the ability to respond to arbitrary utterances since they are limited to a specific domain, thus users have to be guided by the dialog system towards the task at hand. Usually they are deployed to tasks where some information has to be retrieved from a knowledge base. They are mainly employed to replace the process of navigating through menus and user interfaces like making the activity of booking flight tickets or finding public transportation routes between locations conversational [Zhao et al., 2017a].

The second type of dialog agents are the non-task or open-domain chatbots. These conversation systems try to imitate human dialog in all its facets. This means that one should hardly be able to distinguish such a chatbot from a real human, but current models are still far away from such claims. They are usually trained with dialog examples extracted from movie scripts or from Twitter-like post-reply pairs [Vinyals and Le, 2015, Shang et al., 2015, Serban et al., 2016, Li et al., 2016a]. For these models there isn't a well defined goal, but they are required to have a certain amount of world knowledge and commonsense reasoning capabilities in order to hold conversations about any topic.

Recently an emphasis has been put on integrating the two types of conversational agents. The main idea is to combine the positive aspects of both types, like the robust abilities of goal-oriented dialog systems to perform tasks and the human-like chattiness of open-domain chatbots [Zhao et al., 2017a, Yu et al., 2017, Serban et al., 2017b]. This is beneficial because the user is more likely to engage with a task-oriented dialog agent if it's more natural, and handles out of domain responses well.

## 2.2 Early Approaches

ELIZA is one of the first ever chatbot programs written [Weizenbaum, 1966]. It uses clever hand-written templates to generate replies that resemble the user's input utterances. Since then, countless hand-coded, rule-based chatbots have been developed [Wallace, 2009, Carpenter, 2017, Worswick, 2017]. An example can be seen in Figure 1. Furthermore, a number of programming frameworks specifically designed to facilitate building dialog agents have been developed [Marietto et al., 2013, Microsoft, 2017b].

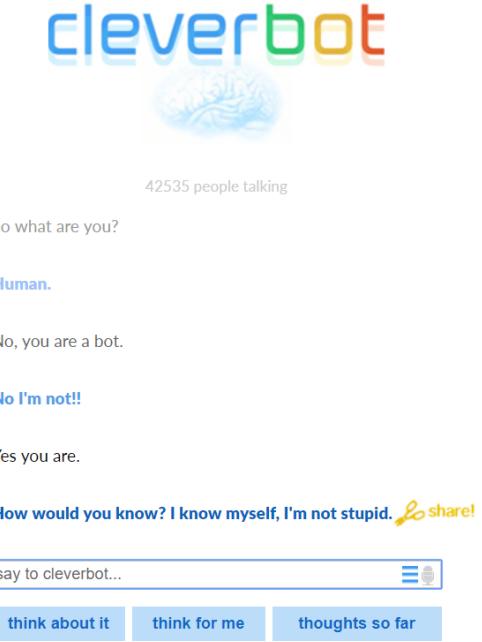


Figure 1: A sample conversation with Cleverbot [Carpenter, 2017]

These chatbot programs are very similar in their core, namely that they all use hand-written rules to generate replies. Usually, simple pattern matching or keyword retrieval techniques are employed to handle the user's input utterances. Then, rules are used to transform a matching pattern or a keyword into a predefined reply. A simple example is shown below in AIML [Marietto et al., 2013]:

```
<category>
  <pattern>What is your name?</pattern>
  <template>My name is Alice</template>
</category >
```

Here if the input sentence matches the sentence written between the *<pattern>* brackets the reply written between the *<template>* brackets is outputted.

Another example is shown below where the *star* symbol is used for replacing words. In this case whatever word follows the word *like* it will be present in the response at the position specified by

the `<star/>` token:

```
<category>
  <pattern>I like *</pattern>
  <template>I too like <star/>. </template>
</category >
```

## 2.3 The Encoder-Decoder Model

The main concept that differentiates rule-based and neural network based approaches is the presence of a learning algorithm in the latter case. An important distinction has to be made between traditional machine learning and deep learning which is a sub-field of the former. In this work, only deep learning methods applied to chatbots are discussed, since neural networks have been the backbone of conversational modeling and traditional machine learning methods are only rarely used as supplementary techniques.

When applying neural networks to natural language processing (NLP) tasks each word (symbol) has to be transformed into a numerical representation [Bengio et al., 2003]. This is done through word embeddings, which represent each word as a fixed size vector of real numbers. Word embeddings are useful because instead of handling words as huge vectors of the size of the vocabulary, they can be represented in much lower dimensions. The vocabulary used in NLP tasks is presented in more detail in Section 2.3.4. Word embeddings are trained on large amounts of natural language data and the goal is to build vector representations that capture the semantic similarity between words. More specifically, because similar context usually is related to similar meaning, words with similar distributions should have similar vector representations. This concept is called the Distributional Hypothesis [Harris, 1954]. Each vector representing a word can be regarded as a set of parameters and these parameters can be jointly learned with the neural network's parameters, or they can be pre-learned, a technique described in Section 3.2.2.

Instead of using hand-written rules deep learning models transform input sentences into replies directly by using matrix multiplications and non-linear functions that contain millions of parameters. Neural network based conversational models can be further divided into two categories, retrieval-based and generative models. The former simply returns a reply from the dataset by computing the most likely response to the current input utterance based on a scoring function, which can be implemented as a neural network [Cho et al., 2014] or by simply computing the cosine similarity between the word embeddings of the input utterances and the candidate replies [Li et al., 2016d]. Generative models on the other hand synthesize the reply one word at a time by computing probabilities over the whole vocabulary [Sutskever et al., 2014, Vinyals and Le, 2015]. There have also been approaches that integrate the two types of dialog systems by comparing a generated reply with a retrieved reply and determining which one is more likely to be a better response [Song et al., 2016].

As with many other applications the field of conversational modeling has been transformed by the rise of deep learning. More specifically the encoder-decoder recurrent neural network (RNN) model (also called seq2seq [Sutskever et al., 2014]) introduced by [Cho et al., 2014] and its vari-

ations have been dominating the field. After giving a detailed introduction to RNNs in Section 2.3.1, the seq2seq model is described in Section 2.3.2. This model was originally developed for neural machine translation (NMT), but it was found to be suitable to *translate* source utterances into responses within a conversational setting [Shang et al., 2015, Vinyals and Le, 2015]. Even though this is a relatively new field, there are already attempts at creating unified dialog platforms for training and evaluating various conversational models [Miller et al., 2017].

### 2.3.1 Recurrent Neural Networks

A recurrent neural network (RNN) [Rumelhart et al., 1988] is a neural network that can take as input a variable length sequence  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$  and produce a sequence of hidden states  $\mathbf{h} = (\mathbf{h}_1, \dots, \mathbf{h}_n)$ , by using recurrence. This is also called the unrolling or unfolding of the network, visualized in Figure 2. At each step the network takes as input  $\mathbf{x}_i$  and  $\mathbf{h}_{i-1}$  and generates a hidden state  $\mathbf{h}_i$ . At each step  $i$ , the hidden state  $\mathbf{h}_i$  is updated by

$$\mathbf{h}_i = f(W\mathbf{h}_{i-1} + U\mathbf{x}_i) \quad (1)$$

where  $W$  and  $U$  are matrices containing the weights (parameters) of the network.  $f$  is a non-linear activation function which can be the hyperbolic tangent function for example. The vanilla implementation of an RNN is rarely used, because it suffers from the vanishing gradient problem which makes it very hard to train [Hochreiter, 1998]. Usually long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997] or gated recurrent units (GRU) [Cho et al., 2014] are used for the activation function. LSTMs were developed to combat the problem of long-term dependencies that vanilla RNNs face. As the number of steps of the unrolling increase it becomes increasingly hard for a simple RNN to learn to remember information seen multiple steps ago.

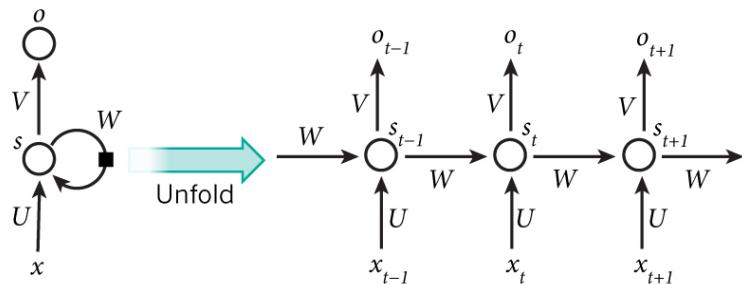


Figure 2: Unfolding of an RNN over 3 time-steps. Here  $x$  is the input sequence,  $o$  is the output sequence,  $s$  is the sequence of hidden states and  $U, W$  and  $V$  are the weights of the network. [Britz, 2015]

LSTM networks use a gating mechanism to control the information flow in the recurrent network. More specifically the input and forgetting gates are used to determine how to update the network's state and the output gate is used to determine what to output from the hidden state.

Mathematically these gates consist of several matrix multiplications and non-linear functions applied to the input vector and the previous hidden state. LSTMs are particularly useful for language modeling, because information has to be preserved over multiple sentences while the network is unrolled over each word. Because of space constraints LSTMs are not detailed further, but a good and quick explanation can be found in [Olah, 2015]. An important characteristic of recurrent neural networks is that the parameters of the function  $f$  don't change during the unrolling of the network.

Language modeling is the task of predicting the next word in a sentence based on the previous words [Bengio et al., 2003]. RNNs can be used for language modeling by training them to learn the probability distribution over the vocabulary  $V$  given an input sequence of words. As previously discussed an RNN receives the word embedding vectors representing words in lower dimensions. The probability distribution can be generated to predict the next word in the sequence by taking the hidden state of the RNN in the last step, and feeding it into a softmax activation function

$$p(\mathbf{x}_{i,j} | \mathbf{x}_{i-1}, \dots, \mathbf{x}_1) = \frac{\exp(\mathbf{v}_j \mathbf{h}_i)}{\sum_{j=1}^K \exp(\mathbf{v}_j \mathbf{h}_i)} \quad (2)$$

for all possible words (symbols)  $j = 1, \dots, K$ , where  $\mathbf{v}_j$  are the rows in the  $V$  weight matrix of the softmax function.  $(\mathbf{x}_{i-1}, \dots, \mathbf{x}_1)$  is the input sequence and  $\mathbf{h}_i$  is the hidden state of the RNN at step  $i$ .

Training of these networks is done via the generalized backpropagation algorithm called truncated backpropagation through time [Werbos, 1990, Rumelhart et al., 1988]. Essentially the error is backpropagated through each time-step of the network to learn its parameters. The error can be computed by using the cross-entropy loss function, which calculates how different the predictions are compared to the true labels.

$$\text{Loss}(\mathbf{y}_i, \hat{\mathbf{y}}_i) = -\mathbf{y}_i \log(\hat{\mathbf{y}}_i) \quad (3)$$

where  $\hat{\mathbf{y}}_i$  is the vector of the predicted probabilities over all words in the vocabulary at step  $i$ , and  $\mathbf{y}_i$  is the one-hot vector over the vocabulary. A one-hot vector is made up of zeros except at the index of the one true word that follows in the sequence, where it is equal to 1. After computing the derivative with respect to all of the weights in the network using the backpropagation through time algorithm, the weights can be updated in order to get closer to an optimum with optimization techniques like stochastic gradient descent (SGD) [Bottou, 2010].

### 2.3.2 The Seq2seq Model

The sequence to sequence model (seq2seq) was first introduced by [Cho et al., 2014], but they only used it to re-rank sentences instead of generating completely new ones, which was first done by [Sutskever et al., 2014]. Since then, besides NMT and conversational models a plethora of applications of these models have been introduced like text summarization [Nallapati et al., 2016], speech recognition [Chiu et al., 2017], code generation [Barone and Sennrich, 2017] and parsing [Konstas et al., 2017], to name a few.

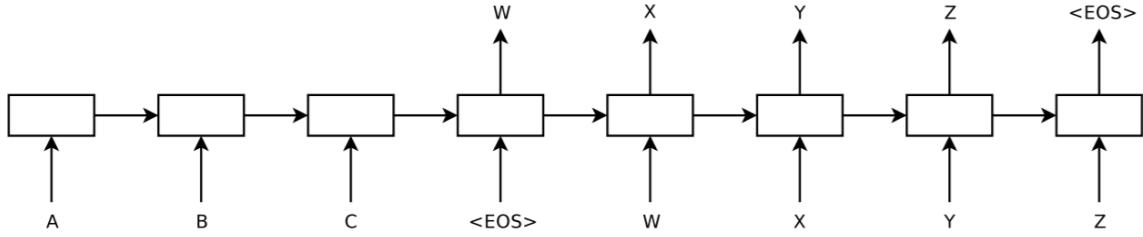


Figure 3: A general seq2seq model, where  $(A, B, C)$  is the input sequence,  $<EOS>$  is a symbol used to delimit the end of the sentence and  $(W, X, Y, Z)$  is the output sequence [Sutskever et al., 2014]

The simplest and initial form of the model is based on two RNNs, visualized in Figure 3. The actual implementation of RNNs can be in the form of LSTMs or GRUs as discussed in Section 2.3.1. The goal is to estimate the conditional probability of  $p(\mathbf{y}_1, \dots, \mathbf{y}_{N'} | \mathbf{x}_1, \dots, \mathbf{x}_N)$ , where  $\mathbf{x}_1, \dots, \mathbf{x}_N$  is the input sequence and  $\mathbf{y}_1, \dots, \mathbf{y}_{N'}$  is the corresponding output sequence. Since two different RNNs are used for the input and output sequences, the lengths of the sequences,  $N$  and  $N'$  can be different. The encoder RNN is unrolled over the words in the source sequence and its last hidden state is called the thought vector, which is a representation of the whole source sequence. The initial hidden state of the decoder RNN is then set to this representation  $\mathbf{v}$ , and the generation of words in the output sequence is done by taking the output of the unrolled decoder network at each time-step and feeding it into a softmax function, which produces the probabilities over all words in the vocabulary:

$$p(\mathbf{y}_1, \dots, \mathbf{y}_{N'} | \mathbf{x}_1, \dots, \mathbf{x}_N) = \prod_{i=1}^{N'} p(\mathbf{y}_i | \mathbf{v}, \mathbf{y}_1, \dots, \mathbf{y}_{i-1}) \quad (4)$$

Training is very similar to a normal RNN, namely the log probability of a correct target sequence  $T$  given the source sequence  $S$  is maximized

$$\frac{1}{S} \sum_{T, S \in S} \log(p(T|S)) \quad (5)$$

where  $S$  is the training set. The two networks are trained jointly, errors are backpropagated through the whole model and the weights are optimized with some kind of optimization technique like SGD.

In NMT the input sequences are sentences in one language from which they have to be translated to the target sequences, which are sentences in a different language. The individual elements of a sequence, or sentence in this case, are vectors representing word embeddings. In conversational modeling the simplest approach is to treat an utterance by a speaker as input sequence and the response to that utterance from a different speaker as the target sequence. Better approaches however are discussed in Section 3.1.1.

### 2.3.3 Deep Seq2seq Models

Seq2seq models can also contain multiple layers of LSTM networks as seen in Figure 4. This is done in order to make the model deeper and to have more parameters, which should ideally lead to better performance [Vinyals and Le, 2015, Wu et al., 2016]. There exist multiple variants, but the most straightforward one is to feed in the source sentence to the first layer of the encoder network. The output from the previous LSTM layer is fed as input to the next layer and the layers are unrolled jointly. Then the last hidden state of the final encoder layer can be used to initialize the initial hidden state of the first decoding layer. The output of the previous decoder layer is input to the next layer until the final layer, where a softmax activation function is applied over the outputs from the last layer to generate the predicted output sequence. How the layers are initialized in the decoder network can be implemented in various ways, like taking the last hidden state from each encoder layer and using it to initialize the first hidden state of each corresponding decoder layer for example.

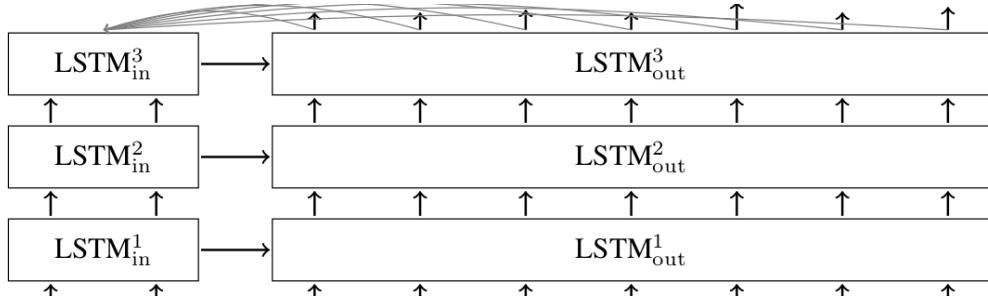


Figure 4: A 3 layer seq2seq model [Tensorflow, 2017]. The lines pointing from the last decoder states to the last encoder represent an attention mechanism, which is presented in Section 3.2.1-

### 2.3.4 Decoding and Vocabulary

In order to get the actual generated output sequences there are several techniques to decode from the probabilities of the words. One such technique is the Roulette Wheel selection, which is commonly used for genetic algorithms [Mitchell, 1998]. Using this method, at each time-step, each word from the vocabulary can be generated with the probability computed from the softmax function. This is useful if the goal is to have some stochasticity in the decoding process, because it can produce slightly different outputs for the same source sequence. However it doesn't perform as well as the more frequent method used, which is to simply output the word with the highest probability from the softmax function. This is a greedy and deterministic approach, since it always outputs the same output for the same input.

While decoding a word at each time-step is fine a better approach is to decode the whole output sequence at the same time, by outputting the sequence with the highest probability.

$$\hat{T} = \arg \max_T p(T|S) \quad (6)$$

Here  $S$  is the source sentence and  $T$  is the target sentence. Since in order to get the sequence with the highest probability, first all of the possible sequences have to be generated, a simple left-to-right beam search is usually employed to make the computation tractable. In the first time-step of the decoder, the top  $K$  words with the highest probabilities are kept. Then at each time-step this list is expanded by computing the joint probability of the partial sequences in the list and the words in the current time-step and retaining the  $K$  most probable partial sequences until the end of the output sequence is reached.

Another important aspect of sequence-to-sequence models applied to tasks involving language is the vocabulary. The vocabulary consists of all the various words and symbols present in the dataset. One problem with this approach is that the vocabulary tends to be quite large, especially for very big datasets like [Lison and Tiedemann, 2016, opensubtitles.org, 2017]. Since the number of parameters of the model increases proportionally with the size of the vocabulary it is usually the case that the vocabulary is limited to some arbitrary size  $N$ . This way only the embeddings of the  $N$  most frequent words in the dataset are used and any other symbols are replaced with a common token representing unknown words. Many approaches have been proposed to the problem of handling out of vocabulary (OOV) or unknown words [Luong et al., 2014, Feng et al., 2017, Jean et al., 2014]. Other methods involve using characters [Zhu et al., 2017] or subword units [Sennrich et al., 2015] instead of words.

### 3 Background

In this section, selected publications are presented first, from the literature research. These papers are grouped into several categories, each corresponding to a specific approach or aspect of conversational modeling. In Section 3.1 further details regarding encoder-decoder models are discussed and in Section 3.2 an overview of several techniques used to augment the performance of encoder-decoder models is given. In Section 3.3 different approaches to conversational modeling are presented, that aren't based on the original seq2seq model.

Finally, in Section 3.4 criticism is presented regarding basic techniques used in neural conversational models. While they are widely used, it is argued that the assumptions on which their usage rests are generally wrong and in consequence these methods are unsuitable for modeling conversations.

#### 3.1 Further Details of Encoder-Decoder Models

In this section further details about seq2seq models are described. First, the context of conversations is described and how context in general can be taken into account in Section 3.1.1. Then, various objective functions that can be used to train seq2seq models are presented in Section 3.1.2. Finally, methods used for evaluating conversational agents are presented in Section 3.1.3.

##### 3.1.1 Context

In this section a commonly used variation of RNNs, called the bidirectional RNN (BiRNN) is described first [Schuster and Paliwal, 1997]. A BiRNN consists of two RNNs, a forward and a backward one, visualized in Figure 5. The forward RNN reads the input sequence in the original order (from  $x_1$  to  $x_n$ ) and computes a sequence of forward hidden states ( $\vec{h}_1, \dots, \vec{h}_n$ ). The backward RNN reads the reversed sequence (from  $x_n$  to  $x_1$ ) and computes the backward hidden states ( $\hat{h}_1, \dots, \hat{h}_n$ ). Then the two hidden states can be simply concatenated for each word in the sequence [Bahdanau et al., 2014, Zhao et al., 2017b]. The BiRNN resembles a continuous bag of words model [Mikolov et al., 2013a], because each concatenated hidden state  $h_i$  has information about the words surrounding the  $i$ -th word. This results in a better preservation of context and is used in several seq2seq models, usually in the encoder network [Zhao et al., 2017b, Xing et al., 2017a, Wu et al., 2016, Yin et al., 2017].

So far, only conversational modeling as predicting a reply based on a single source utterance has been discussed. However, conversations are more complicated than machine translation, since they don't solely consist of utterance-reply pairs. Conversations usually have many turns, and the reply to an utterance might depend on information presented in previous turns. A plethora of approaches have been proposed to incorporate context or conversation history into seq2seq models in order to build better dialog agents. Perhaps the most straightforward approach is to concatenate  $k$  previous utterances by appending an end-of-utterance symbol after each utterance and feeding this long sequence of symbols into the encoder [Vinyals and Le, 2015]. The simplest approach which was used as a baseline in [Sordoni et al., 2015] is to use only the first preceding utterance

as context in order to form context-message-reply triples for training. A better approach was to concatenate the bag of words representations of the context and message utterances instead of the actual utterances. By using different representations for different utterances better results were achieved [Sordoni et al., 2015].

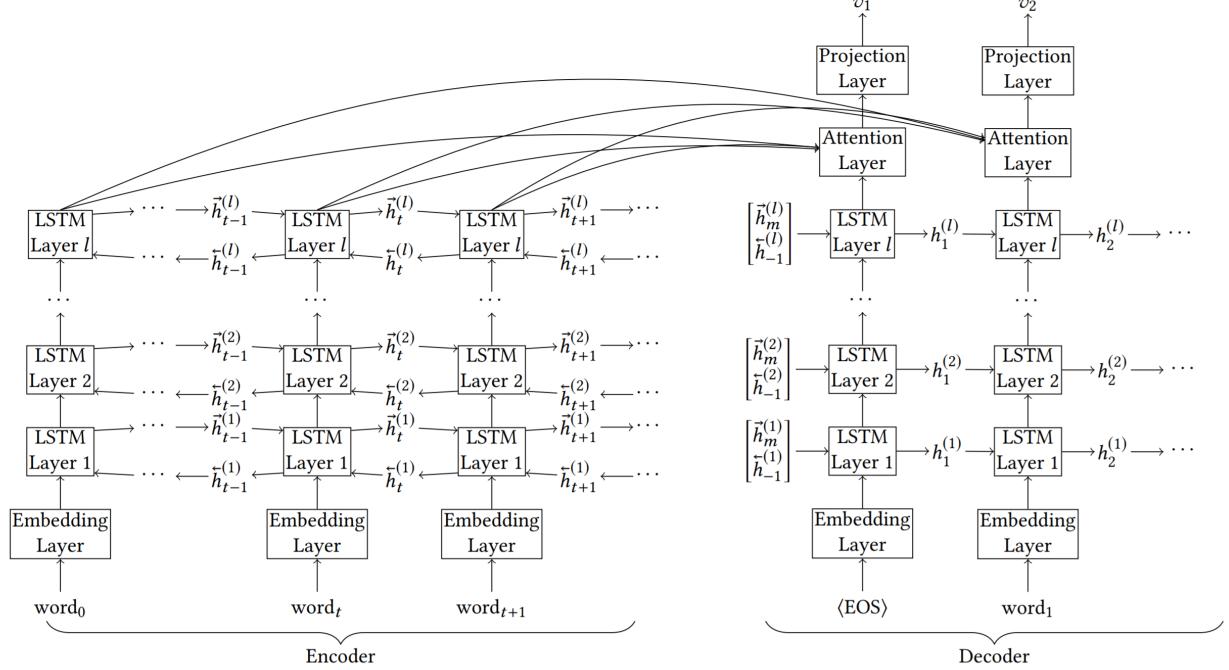


Figure 5: A bidirectional multilayer LSTM encoder and unidirectional LSTM decoder [Yin et al., 2017] with attention. Attention is described in Section 3.2.1.

However feeding long sequences of symbols into the encoder RNN of seq2seq models accentuates the vanishing gradient problem [Hochreiter, 1998]. Moreover, as sequences get longer it becomes increasingly more difficult for RNNs to retain information which was inputted many time-steps ago. Consequently, it becomes even harder to encode all relevant information into the last hidden state of the encoder network from a sequence consisting of multiple sentences.

A possible solution is to represent the current utterance and previous dialog turns with different RNNs and to build a hierarchical representation of the conversation history [Serban et al., 2016], described in detail in Section 3.3.1. A similar approach was presented in [Zhao et al., 2017b] using one RNN to encode the current utterance and a different RNN to encode  $k$  previous utterances, but this approach doesn't take into consideration the hierarchical nature of dialogs.

### 3.1.2 Objective Functions

In addition to the standard cross-entropy loss introduced in Section 2.3.1, various loss functions have been proposed for conversational models in order to generate more varied and interesting

replies. One such class of loss functions are reinforcement learning based, discussed in detail in Section 3.3.3. In [Ramachandran et al., 2016] a seq2seq model is pretrained as a language model and in order to avoid overfitting, monolingual language modeling losses are added to the standard loss function. These losses act as a regularizer by forcing the seq2seq model to correctly model both the normal seq2seq task and the original monolingual language modeling tasks. Pretraining is presented in more detail in Section 3.2.2. In [Lison and Bibauw, 2017] it is argued that not all context-response pairs in a dialog dataset have the same importance for conversational modeling. Accordingly, each context-response pair is weighted with a scoring model and these weights are used in the loss function when training a seq2seq model. Hence, examples associated with bigger weights in the dataset will have a larger impact on the gradient update steps through the loss function.

Approaches to incorporate beam search into the training procedure have also been explored. In [Wiseman and Rush, 2016] a simple approach is taken to construct the loss function by penalizing when the gold target sequence is not present in the beam consisting of the most likely predictions. The issue with incorporating beam search directly into the training procedure of seq2seq models is that it uses the argmax function which is not differentiable and hence is not amenable to back-propagation. Since the predictions, which serve as input to the standard loss function are discrete (from beam search), the evaluation of the final loss is also discontinuous. A relaxation based technique is proposed in [Goyal et al., 2017], where the standard loss function is gradually relaxed to the beam search based loss function as training progresses.

Perhaps the most extensively used objective function besides the standard one is based on Maximum Mutual Information (MMI) [Li et al., 2015]. In MMI the goal is defined as maximizing pairwise or mutual information between a source sentence  $S$  and the corresponding target  $T$ :

$$\log \frac{p(S, T)}{p(S)p(T)} = \log p(T|S) - \log p(T) \quad (7)$$

By introducing a weighting constant  $\lambda$  and by using the Bayes' theorem the training objective can be written in the following two ways:

$$\hat{T} = \arg \max_T [\log p(T|S) - \lambda \log p(T)] \quad (8)$$

$$\hat{T} = \arg \max_T [(1 - \lambda) \log p(T|S) + \lambda \log p(S|T)] \quad (9)$$

However, it is not trivial to use these equations as loss functions in practice. In Equation 8 the term  $\lambda \log p(T)$  acts as an anti-language model since it is subtracted from the loss function. This leads to ungrammatical output sentences in practice. Training using Equation 9 is possible, since the only requirement is to train an additional seq2seq model for the  $\log p(S|T)$  term by using targets as inputs and inputs as targets, but direct decoding is intractable since it requires completion of target generation before  $p(S|T)$  can actually be computed. Solutions to these issues have been proposed in [Li et al., 2015].

### 3.1.3 Evaluation Methods

Evaluation of dialog systems is perhaps a more controversial topic. Indeed it is still an open research problem to find good automatic evaluation metrics that can effectively compare the performance of conversational models. The traditional approach is to use the same metrics that are used for NMT and language modeling. Bleu and perplexity are widely used metrics for evaluating dialog systems [Vinyals and Le, 2015, Yao et al., 2016, Zhao et al., 2017a, Serban et al., 2016].

Bleu [Papineni et al., 2002] measures how many n-word-sequence (n is usually 4) overlaps are there between the test sentences and the predicted ones. Similar to this, but less commonly used is the simple accuracy of the whole sentence or the word error rate, which is prevalent for speech recognition models [Shannon, 2017, Park et al., 2008]. It calculates how many words are incorrect in the predicted sentence.

Perplexity [Manning et al., 1999] is another common metric, which measures how well a probability model predicts a sample. Given a model, sentences can be inputted that weren't used during training and the perplexity on the test set can be computed. If a model  $q$  exists (encoder-decoder for conversational modeling), the perplexity can be computed by

$$2^{-\frac{1}{N} \sum_{i=1}^N \log_2 q(\mathbf{x}_i)} \quad (10)$$

where  $\mathbf{x}_i$  are the test samples or words in the sentence and  $N$  is the length of the sentence. The goal is for models to assign very high probability to test samples, meaning that the lower the perplexity score the better.

Unfortunately it has been shown that these metrics don't correlate well with human judgment in the conversational setting [Liu et al., 2016]. The downfalls of these standard metrics are discussed in Section 3.4.4. Various methods have been proposed to address the problem of evaluating conversational agents, for example how many turns it takes until the model produces a generic answer [Zhao et al., 2017a, Li et al., 2016c] or the diversity of the generated responses [Li et al., 2016c]. More sophisticated metrics make use of neural network based models to assign a score to utterance-response pairs and it has been shown that they correlate better with human judgments than traditional metrics [Lowe et al., 2017, Tao et al., 2017].

For reinforcement learning and task-oriented dialog agents, described in Section 3.3.3 and Section 3.3.2 respectively, evaluation is more straightforward. In reinforcement learning a goal that the dialog agent has to achieve is specified, and its performance can be simply measured based on the percent of cases in which it achieves the goal [Li et al., 2017, Havrylov and Titov, 2017]. Similarly for task-oriented dialog agents usually there is a clearly defined task and the accuracy of accomplishing the given task serves as a good performance metric [Joshi et al., 2017, Zhao et al., 2017a, Li et al., 2016b].

Finally, one of the better methods to evaluate dialog systems is to ask other people what they think about the quality of the responses that an agent produces. Human judgment has been one of the most prevalent metrics throughout recent literature related to conversational modeling [Shang et al., 2015, Vinyals and Le, 2015, Zhou et al., 2017, Li et al., 2016c, Zhao et al., 2017a, Li et al., 2016c, Li et al., 2015]. Usually human judges are asked to either rate the quality of individual responses from a model on a scale or to choose the better response from responses produced by

different models for the same input. A number of properties of conversations can be rated, like naturalness, grammaticality, engagement or simply the overall quality of the dialog.

## 3.2 Augmentations To The Encoder-Decoder Model

In this section recent methods and techniques used to augment the performance of encoder-decoder models are presented. In Section 3.2.1 various attention mechanisms are discussed. Furthermore, it is shown how pretraining of seq2seq models can help in a conversational setting in Section 3.2.2. Then, various features and priors that can be used as additional inputs to seq2seq models are presented in Section 3.2.3. Finally, it is shown how knowledge bases and information retrieval techniques can be good additions to encoder-decoder models in Section 3.2.4.

### 3.2.1 Attention

The attention mechanism visualized in Figure 6 was first introduced to encoder-decoder models by [Bahdanau et al., 2014]. The problem that it was trying to address is the limited information that the context vector can carry. In a standard seq2seq model it is expected that this single fixed-size vector can encode all the relevant information about the source sentence that the decoder needs in order to generate its prediction. Additionally, since only a single vector is used, all the decoder states receive the same information, instead of feeding in information relevant to the specific decoding step.

In order to combat these shortcomings the attention mechanism creates an additional input at each decoding step coming directly from the encoder states. This additional input  $\mathbf{c}_i$  to the decoder at time-step  $i$  is computed by taking a weighted sum over the encoder hidden states  $\mathbf{h}$ :

$$\mathbf{c}_i = \sum_{j=1}^T a_{ij} \mathbf{h}_j \quad (11)$$

where  $T$  is the number of hidden states or symbols in the source sentence. The weight  $a_{ij}$  for each hidden state  $\mathbf{h}_j$  can be computed by

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^T \exp(e_{ik})} \quad (12)$$

which is basically a softmax function over  $e_{ij}$ , which is the output of a scoring function:

$$e_{ij} = f(\mathbf{s}_{i-1}, \mathbf{h}_j). \quad (13)$$

Here  $\mathbf{s}_{i-1}$  is the hidden state of the decoder in the previous time-step. Oftentimes  $\mathbf{s}_{i-1}$  is called a query vector ( $\mathbf{q}$ ) and the encoder hidden states  $\mathbf{h}$  are called key vectors ( $\mathbf{k}$ ). A good visualization of the weights  $a_{ij}$  between the source and output sentence can be seen in Figure 7. The scoring function  $f$  can be implemented in several ways. In Equation 14 a Multi-Layer Perceptron (MLP) approach can be seen, where the query and key vectors are simply concatenated and fed into a

feed-forward neural network [Bahdanau et al., 2014]. The Bilinear (BL) scoring function is described in Equation 15, proposed by [Luong et al., 2015]. The Dot Product (DP) scoring function (Equation 16) doesn't use any weights, but it requires the sizes of the two vectors to be the same [Luong et al., 2015]. An extension of this is the Scaled Dot Product (SDP) function (Equation 17), where the dot product is scaled by the square root of the size of the key vectors [Vaswani et al., 2017]. This is useful because otherwise the dot product increases as dimensions get larger.

$$\text{MLP}(\mathbf{q}, \mathbf{k}) = \mathbf{w}_2 \tanh(W_1[\mathbf{q} : \mathbf{k}]) \quad (14)$$

$$\text{BL}(\mathbf{q}, \mathbf{k}) = \mathbf{q}^\top W \mathbf{k} \quad (15)$$

$$\text{DP}(\mathbf{q}, \mathbf{k}) = \mathbf{q}^\top \mathbf{k} \quad (16)$$

$$\text{SDP}(\mathbf{q}, \mathbf{k}) = \frac{\mathbf{q}^\top \mathbf{k}}{\sqrt{|\mathbf{k}|}} \quad (17)$$

In the above equations  $W$ ,  $W_1$  and  $\mathbf{w}_2$  are parameters of the scoring functions.

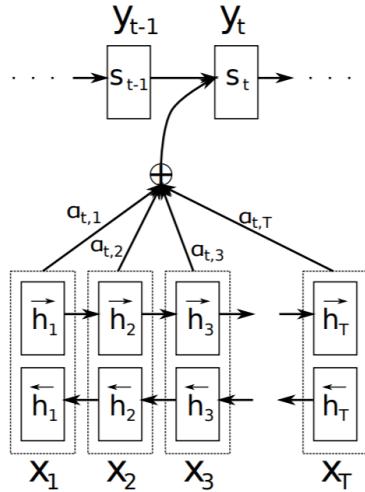


Figure 6: Original Attention Mechanism [Bahdanau et al., 2014]

Outputs from the decoder network generated in previous time-steps can be attended to by incorporating them as additional inputs to the scoring functions mentioned above [Shao et al., 2017]. This helps the decoder by allowing it to *see* what has already been generated.

Attention can also be implemented using multiple heads [Vaswani et al., 2017], meaning that the output of multiple scoring functions is used, each with their own parameters, in order to learn to focus on different parts of the input sequence. The parameters of the scoring functions are jointly learned with all other parameters of the seq2seq model.

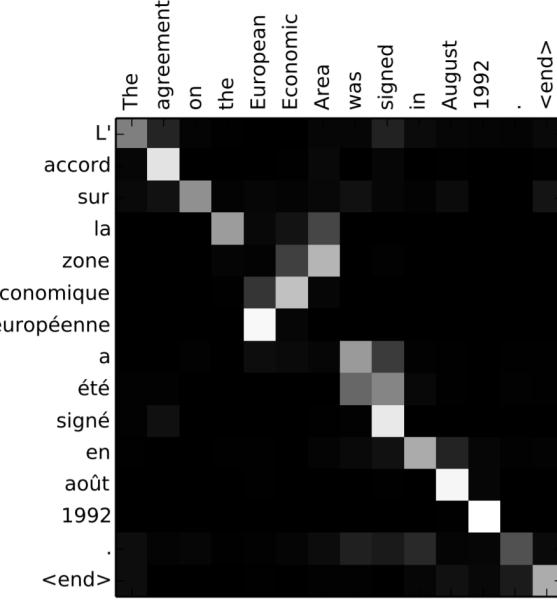


Figure 7: Pixels represent the weights  $a_{ij}$  between inputs and outputs [Bahdanau et al., 2014]

Attention can also be computed using only one sequence of symbols, called self- or intra-attention. Self-attention has been successfully applied to a variety of tasks including reading comprehension, abstractive summarization, neural machine translation and sentence representations [Cheng et al., 2016, Lin et al., 2017, Vaswani et al., 2017, Parikh et al., 2016, Paulus et al., 2017]. In this case the query and key vectors both come from the same hidden states and thus an encoding of the sentence which depends on all of its parts is achieved. It can be used both in the decoder and the encoder networks since it can be implemented as a standalone layer that takes in a sequence and computes a new representation of that sequence. Figure 8 depicts the self-attention mechanism between the same sentence.

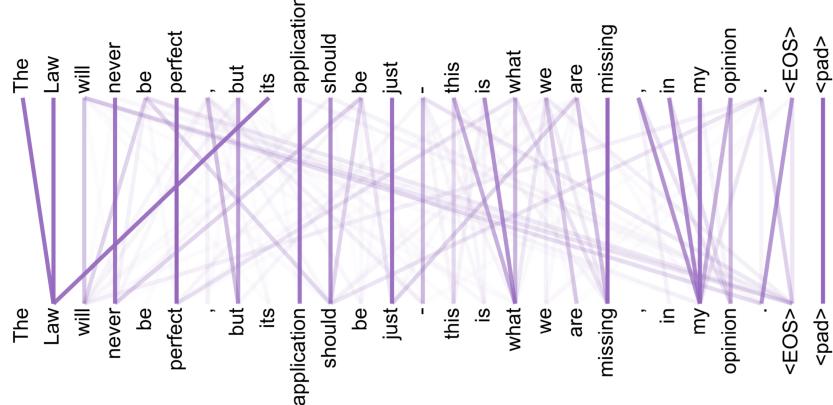


Figure 8: Visualizing self-attention, lines represent the weights  $a_{ij}$  [Vaswani et al., 2017]

Attention has been used extensively for neural conversational models as an augmentation of the base seq2seq model [Yao et al., 2016, Shang et al., 2015, Xing et al., 2017a, Zhao et al., 2017a]. A more complex type of attention mechanism is the hierarchical attention which was used in conversational modeling [Xing et al., 2017b] and abstractive text summarization [Nallapati et al., 2016] as well. It is useful when it is necessary to attend over multiple input sentences, described in more detail in Section 3.3.1.

### 3.2.2 Pretraining

Pretraining for seq2seq models means that instead of initializing the parameters of a model randomly the model is first pretrained on some data or some other task that is different from the main task that the model needs to be applied to. One of the most common approaches among many NLP tasks is to pretrain the parameters of the word embeddings [Chen and Manning, 2014, Serban et al., 2016, Akasaki and Kaji, 2017, Lample et al., 2016, Serban et al., 2017b]. The most popular techniques to pretrain word embeddings are presented in [Mikolov et al., 2013a, Mikolov et al., 2013b]. An advantage of pretraining word embeddings is that during the actual training of the seq2seq model these embeddings can be fixed and thus the model has to learn less parameters.

In addition to pretraining word embeddings, all of the parameters of an encoder-decoder model can be pretrained as well. For conversational modeling this is very beneficial, because oftentimes well labeled datasets are relatively small. By pretraining on a big but noisier dataset the parameters of a model, it will already achieve somewhat good performance. Thus the model will have learned a good amount of general knowledge about the task [Li et al., 2016a, Serban et al., 2016]. Then, by finetuning on a smaller dataset it can achieve better performance without overfitting. For example a conversational model can learn general knowledge like answering with yes/no to a yes-or-no question. Then, during finetuning the model doesn't have to learn all of this knowledge again and thus can focus on more subtle properties of conversations or domain-specific knowledge. In [Li et al., 2016a] a seq2seq model was pretrained on the OpenSubtitles dataset [Lison and Tiedemann, 2016] and then the pretrained model was adapted to a much smaller TV series dataset. A somewhat different approach was employed in [Ramachandran et al., 2016], where the authors pretrained the encoder and decoder RNNs of a seq2seq model separately as language models. Thus both networks already had some knowledge about language before putting them together and training the whole seq2seq model for NMT. Similarly in [Sriram et al., 2017] a pretrained language model (LM) was used to augment the performance of a standard seq2seq model. With the information from the LM the seq2seq model was able to improve its performance for domain adaptation, which means that it performed almost as good on a different domain as on the domain it was trained on.

In [Lowe et al., 2017] the authors made an attempt at building a model which automatically assigns scores to sample conversations. To do this they used the encoder and context RNN part of a HRED network (described in Section 3.3.1) to encode the conversation. Since their dataset of scored conversations was small they resorted to pretrain the HRED with the normal task of generating replies to conversations.

Lastly, a different style of pretraining was employed in [Wiseman and Rush, 2016]. Instead of pretraining a seq2seq model with a different dataset they pretrained the model with a different loss

function. This was necessary since the authors tried to integrate beam search into the loss function, but found that without first pretraining with the standard cross-entropy loss function the model was unable to learn anything with their new loss function.

### 3.2.3 Additional Input Features

In addition to the raw dialog turns a plethora of other inputs can be integrated into the seq2seq model. A number of attempts have been made since the birth of the encoder-decoder model in order to augment it with additional input features. The goal of these features is mainly to provide more information about the conversation or the context. In addition, models infused with additional features can learn to differentiate between various styles of dialog. For example, by conditioning a seq2seq model on speaker information it can learn to output replies in the style of the speaker it was trained on [Li et al., 2016a]. In this paper the authors input additional speaker embeddings into the decoder. These speaker embeddings are similar to word embeddings, basically representing the speakers for the utterances in the dataset. The additional speaker embedding vector is fed into the decoder RNN at each time-step jointly with the previously generated word. The result is a more consistent conversational model, visualized in Figure 9. Because of the speaker embeddings it learns general facts associated with the specific speaker. For example to the question *Where do you live?* it might reply with different answers depending on the speaker embedding that is fed into the decoder.

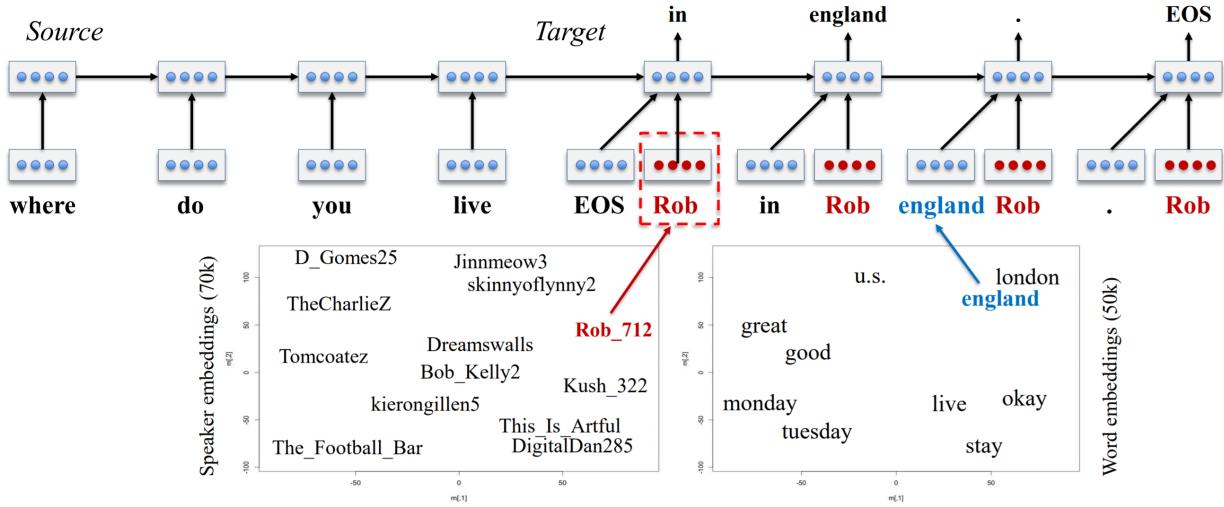


Figure 9: A seq2seq model augmented with speaker embeddings [Li et al., 2016a].

Additionally the authors have also experimented with speaker-addressee models. This is a simple extension to the previously discussed speaker model. Instead of simply feeding in speaker embeddings to the decoder RNN, a weighted sum of the speaker and addressee embeddings is inputted. The addressee is the speaker to which the utterance is directed. The intuition behind this

addition is that people talk differently depending on who they talk to (boss, friend, lover). Thus, the model not only responds differently with different speaker embeddings but it also conditions its response on the addressee embeddings. Hence, it might output a different reply for the same utterance and speaker embedding, but different addressee embedding.

Similarly, many other approaches have been made to condition the response generation of seq2seq models on various types of categories. For example in [Xing et al., 2017a] topic words are extracted from each utterance in the dataset. Then, a seq2seq model is trained with these additional topic words together with the utterance they belong to. More specifically, a separate attention mechanism is implemented that attends over the topic words. The two vectors generated from the normal utterance attention and the topic attention are fed into the decoder RNN at each time-step. With this additional topic information the model manages to produce more relevant replies to the topic of the conversation. A somewhat different approach to include topic information into the response generation process was employed in [Choudhary et al., 2017]. In this work three seq2seq models were trained separately on three datasets related to different topics (domains). Additionally, a domain classifier based on logistic regression was trained to compute the probability that the utterance is related to a domain, for all three domains. At test time, for an input utterance all the seq2seq models generate a reply and the domain classifier predicts a domain. Then, a re-ranker takes the generated responses and the predicted domain probabilities and based on their product, determines the most probable reply-domain pair. Another example involves using emotion categories for post-response pairs [Zhou et al., 2017]. Here the authors first classify all of the post-response pairs in the dataset with an LSTM model into several emotion categories like happy, sad, angry, etc. Then, they feed in these categories as real-valued vector representations into the decoder of an encoder-decoder model during training. Additionally, a more complex memory based augmentation is proposed to the encoder-decoder model to better capture emotional changes in the response, which is not detailed here. In essence, by feeding in these emotion categories a reply conditioned on a specific type of emotion can be generated. For example, the response for the question *How was your day?* will differ in style for different emotion categories.

A different approach is taken in [Ghazvininejad et al., 2017], where the emphasis is put on taking into account relevant facts to an utterance. A seq2seq model is upgraded with a fact encoder operating over a knowledge base, which stores facts about various locations (eg. restaurants, theaters, airports). Before generating the reply to an encoded utterance, a location or entity is extracted from it with keyword matching or named entity recognition. Then, based on the location or entity, relevant facts are selected from the knowledge base, which are encoded by a memory network [Sukhbaatar et al., 2015]. Afterwards the vector representation from the encoded facts and the vector representation from the encoded utterance are summed and fed into the decoder RNN of the encoder-decoder model which generates the response. The authors used Twitter post-reply-reply 3-turn conversations which included a handle or hashtag about locations for which relevant facts were selected from a knowledge-base. With this approach the conversational model managed to produce more diverse and relevant replies if there was a location identified in the source utterance. Similar methods involve leveraging information from knowledge bases as additional inputs, described in detail in Section 3.2.4.

Other approaches try to integrate more features into the seq2seq model without using additional

information. In a standard seq2seq model the only information the model has about the source utterance is through the vector representations of the words. To enrich the representation of the natural language utterances many other features have been proposed to be fed into the encoder RNN together with the word embeddings [Sordoni et al., 2015, Serban et al., 2017a, Serban et al., 2017b]. Such features include part-of-speech tags, dialog acts and n-gram word overlaps, to name a few.

### 3.2.4 Knowledge Bases and Copying

Knowledge bases (KB) are powerful tools that can be used to augment conversational models. Since knowledge bases usually entail some kind of domain specific information, these techniques are mainly used for task-oriented dialog systems, presented in detail Section 3.3.2. In a KB, information related to the task at hand can be stored, for example information about nearby restaurants or about public transportation routes. Simple dictionaries or look-up-tables can be used to match an entity with information about it. Since KBs store information discretely, their integration with neural network based encoder-decoder models is not trivial.

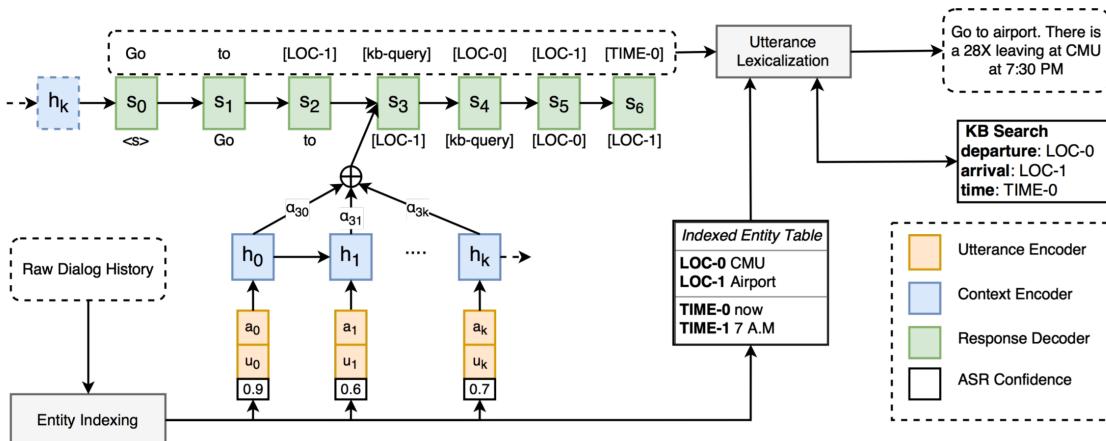


Figure 10: A HRED with attention, described in detail in Section 3.3.1, upgraded with a KB component [Zhao et al., 2017a].

Perhaps the most straightforward integration method is proposed in [Zhao et al., 2017a]. In this work an entity indexing step is introduced before feeding in the source utterance into a seq2seq model. This works by replacing locations or time specifying words with general tokens using named entity recognition or keyword matching. For example in the sentence *I want to go to Paris from New York*, the word *Paris* is replaced with the token *[LOC-1]* and *New York* is replaced with *[LOC-0]*. The connection between the general token and the original word is stored in a table. Then, the encoder-decoder model produces a response that also uses general tokens for locations and times, and a special placeholder token for the KB result. Finally, the general tokens are transformed back to actual words using the stored table, a KB is employed which uses these general

tokens to search for a route between the two places and its output is incorporated in the response. Thus, an open-domain dialog system is achieved, which is augmented with a task-oriented, robust feature handling user requests related to finding routes between two locations. A visualization of the architecture of this model can be seen in Figure 10.

A similar, but more complex approach is taken in [Wen et al., 2016], where a KB augmented encoder-decoder model is used for the task of recommending restaurants. Here, besides a standard encoder RNN the source utterance is also processed with a belief tracker, implemented as a convolutional neural network (CNN). Convolutional neural networks applied to encoder-decoder models are discussed in more detail in Section 3.3.4. Belief tracking is an important part of task-oriented spoken dialog systems [Henderson, 2015]. The belief tracker network produces a query for a MySQL database containing information about restaurants. The final input to the decoder RNN is the weighted sum consisting of the last state of the decoder RNN and a categorical probability vector from the belief tracker. Then the decoder outputs a response in the same way as in the previous example, with lexicalised general tokens. These tokens are then replaced with the actual information that they point to in the KB. Similarly, in [Ghazvininejad et al., 2017] a more general fact based KB is employed to augment an encoder-decoder model, which is presented in detail in Section 3.2.3.

One of the most popular task-oriented datasets is also in the restaurant domain [Joshi et al., 2017]. Here special API calls are implemented to search over a KB. In order to call the API functions the dialog system has to first identify all the inputs by retrieving them from the user’s utterances (eg. type of restaurant, price range). This task is described in detail in Section 3.3.2. A more general attempt to retrieve relevant information by the use of queries based on information probabilities is presented in [Yin et al., 2017].

Named entity detection is also used together with a KB to solve a different problem in conversational modeling [Li et al., 2016d]. In this work the authors try to solve the problem of users getting disinterested in the conversation with a dialog agent. When the model detects that the user is bored (eg. writes ...) the normal dialog system is replaced with a content introducing mechanism. More specifically, the last couple of utterances are searched to retrieve any named entities. Then, these entities are inputted to a KB, which contains associations between various entities, for example between a movie title and actor names in the movie. This information is then used to produce the response. Hence, the dialog system can introduce new and relevant content based on a KB.

Another interesting line of research in conversational modeling is the integration of copying mechanisms in order to deal with out-of-vocabulary words [Eric and Manning, 2017, Gu et al., 2016]. In [Gu et al., 2016] the probability of generating a word  $y_t$  at time-step  $t$  is given by summing the probabilities from generate- and copy-modes. These probabilities are given by scoring functions. The scoring function of the generate-mode produces a score for each word in the vocabulary based on simple multiplication of the current decoder hidden state with a learned weight matrix. In contrast the scoring function for copy-mode produces scores for each OOV word  $x_i$  in the source sentence. This scoring function is very similar to an attention mechanism:

$$f(x_i) = \sigma(\mathbf{h}_i^\top W_c) s_t \quad (18)$$

where  $\mathbf{h}_i^\top$  is the hidden state of the encoder at step  $i$ ,  $s_t$  is the hidden state of the decoder at the

current time-step,  $W_c$  is a learned weight matrix and  $\sigma$  is a non-linear activation function. Using this copy-mode the model is able to efficiently handle OOV words and even integrate them into the response, making replies more diverse and unique.

### 3.3 Different Approaches to Conversational Modeling

In this section hierarchical models used for building conversational agents are presented in Section 3.3.1. Then, various approaches to integrate task-oriented conversations and goals with encoder-decoder models are discussed in Section 3.3.2. Furthermore, reinforcement learning based approaches, that have seen some success when applied to the task of training conversational agents recently, are presented in Section 3.3.3. Finally, encoder-decoder models that are very different from a standard RNN based seq2seq, but nonetheless have achieved state-of-the-art results, are described in Section 3.3.4.

#### 3.3.1 Hierarchical Models

In order to better represent dialog history, a general hierarchical recurrent encoder-decoder (HRED) architecture was proposed in [Serban et al., 2016]. The model consists of three different RNNs, the encoder RNN, the context RNN and the decoder RNN. First,  $k$  previous utterances of a conversation are encoded separately by the encoder RNN. This produces  $k$  separate context vectors by taking the last hidden state of the encoder RNN from each encoded utterance. Then these  $k$  hidden states are fed into the context RNN step by step, thus it has to be unrolled for  $k$  steps. Next, the last hidden state from the context is used to initialize the decoder RNN. The decoder RNN and the decoding process is very similar to the one found in a normal seq2seq model.

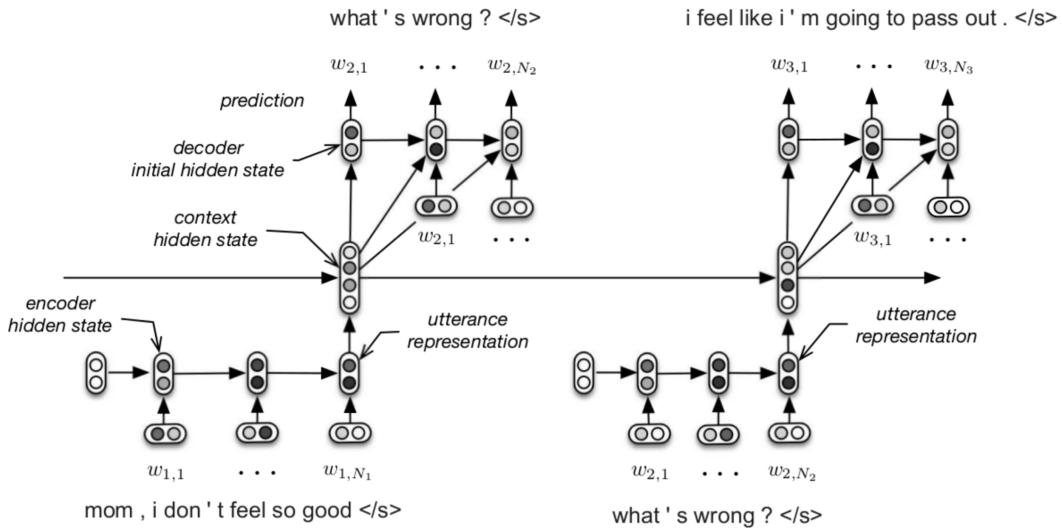


Figure 11: The Hierarchical Recurrent Encoder-Decoder Model [Serban et al., 2016].

The HRED (Figure 11) differs from the basic encoder-decoder model by introducing a new level of hierarchy, the context RNN, which computes hidden states over entire utterances. With this approach the natural hierarchy of conversations is preserved by handling word sequences with the encoder RNN and utterance sequences with the context RNN which depends on the hidden representations produced by the word-level encodings.

Since the introduction of the HRED model a number of works have used and augmented it [Serban et al., 2017c, Serban et al., 2017a, Serban et al., 2017b, Shen et al., 2017, Li et al., 2017]. A proposed extension to the HRED model is to condition the decoder RNN on a latent variable, sampled from the prior distribution at test time and the approximate posterior distribution at training time [Serban et al., 2017c]. These distributions can be represented as a function of previous subsequences of words.

In [Serban et al., 2017a] two HRED models are employed simultaneously. One HRED operates over coarse tokens of the conversation (eg. POS tags) to generate coarse predictions. Then, the second HRED, which takes as input natural language utterances, generates natural language predictions by conditioning on the predicted coarse tokens. This conditioning is done via concatenation of the last hidden state of the decoder RNN of the coarse predictions with the current context RNN hidden state from the natural language HRED and feeding it into the natural language decoder RNN.

In order to handle the two-party style of conversations, two separate hierarchical recurrent encoder (HRE) models are used in [Shen et al., 2017]. The HRE is the same as the HRED without the decoder RNN. One of the HRE networks encodes only the utterances coming from one of the speakers, and the other HRE encodes the utterances of the other speaker. Thus, at each turn the two networks produce two hidden context states which are concatenated and fed into a single decoder RNN, producing the output prediction.

A natural extension to the original HRED model is to incorporate attention mechanisms, explored in several works [Yao et al., 2015, Yao et al., 2016, Xing et al., 2017b]. The simplest form of integrating attention into the HRED model is between the previous decoder RNN hidden state and the encoder RNN hidden states from the previous turn. This can be done in exactly the same way as in standard seq2seq models and has been explored in [Yao et al., 2015, Yao et al., 2016].

A more interesting approach is to make use of the hierarchical structure of the HRED and integrate attention hierarchically as well [Xing et al., 2017b]. Accordingly, the model in this work is called the hierarchical recurrent attention network (HRAN). There are two levels of attention employed. At each time-step, the word level attention mechanism computes vector representations over the hidden states of the encoder RNN (keys) and the previous hidden state of the decoder RNN (queries). Each utterance is encoded separately by the encoder RNN and word level attention is also computed separately for each utterance. The produced vector representations are fed into the utterance level encoder (context RNN). Then, the utterance level attention mechanism computes a context vector based on the hidden states of the context RNN. The last step is to feed this context vector into the decoder RNN at each step, which computes the output sequence predictions.

An additional technique employed in the HRAN architecture is to use context RNN hidden states as inputs to the word level attention. More specifically the context RNN is implemented in a reverse order, meaning that if there is a sequence of utterances  $(u_1, \dots, u_n)$ , the encoder RNN

first encodes  $u_n$ , thus the utterance level encoder will also start with the vector representation from the last utterance in the conversation. Because the utterances are reversed, the word level attention mechanism for each utterance can use as additional input the hidden state of the context RNN from the next utterance in the original order. Figure 12 depicts the HRAN with all the components mentioned above.

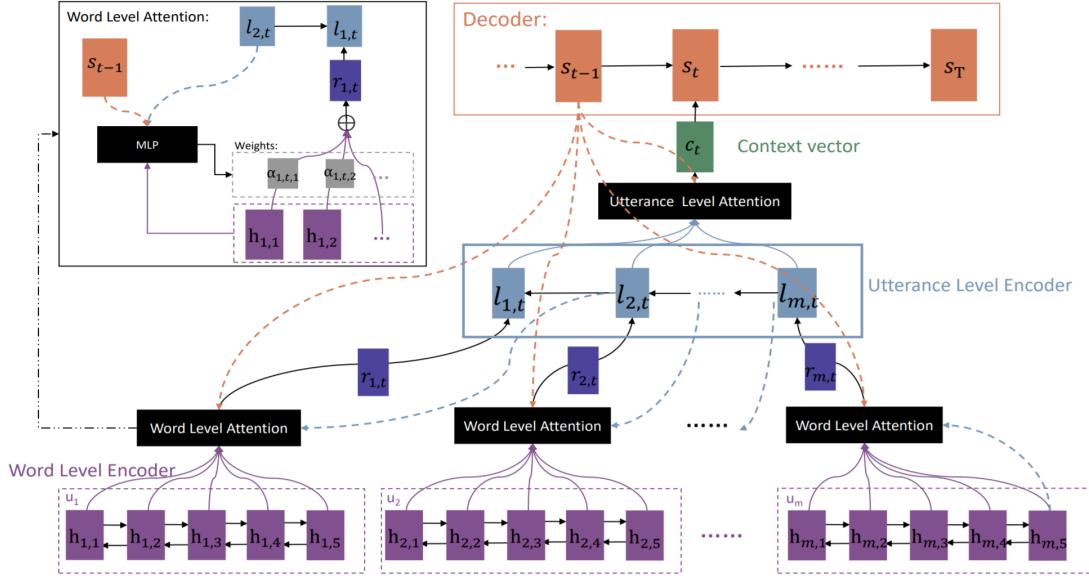


Figure 12: The Hierarchical Recurrent Attention Network [Xing et al., 2017b]

### 3.3.2 Task-Oriented Dialog Systems

As it was discussed in Section 2.1 task-oriented dialog agents are a sub-type of conversational models. They are more narrow with regard to language understanding and conversation topics, but they make up with robustness, making them better suited for commercial deployment. In task-oriented dialog systems there is usually a general task defined, and a goal to be achieved through conversation. A common property of task-oriented systems is that they learn from smaller and task-specific datasets. Since it is usually not required to generalize to other domains, learning on smaller, but more focused datasets gives much better performance. Furthermore, KB components described in Section 3.2.4 are often used in combination with neural network models. KBs are an essential component of task-oriented dialog systems since they provide the model with accurate and rich information. Thus, neural network models only have to learn how to carry general conversations, how to form queries to KBs and how to incorporate the result from KB look-ups. Also, rule-based systems often perform equally well at task-oriented conversations, since only certain types of questions related to the tasks are expected from the user.

A prevalent task and goal is the recommendation of restaurants to users searching for specific types of cuisine. One of the most popular benchmark tasks is situated in this domain, named bAbi

[Bordes et al., 2016, Joshi et al., 2017, Facebook, 2017]. Since this is a simulated dataset, a rule-based system can achieve 100% task success rate. Nonetheless it is an important benchmark for neural conversational models. The task is further divided into 5 sub-tasks to measure the performance on individual parts of the dialog. Task 1 consists of dialog conducted until the first API call. Here the dialog agent has to gather sufficient information from the user’s utterances in order to call an API function. This function usually takes as input the type and location of the restaurant, the number of people for which the reservation should be made and the price range. Based on these inputs it automatically returns a list of candidate restaurants using a KB. Task 2 measures the ability of the dialog agent to handle instances when the user forgot to mention an option or would like to change one. Then, in Task 3 the agent presents a possible candidate and converses with the user until the final restaurant is selected by the user. Finally, Task 4 measures the ability of the dialog agent to provide additional information about the restaurant like the address or the phone number. Task 5 measures the overall success of the full dialog.

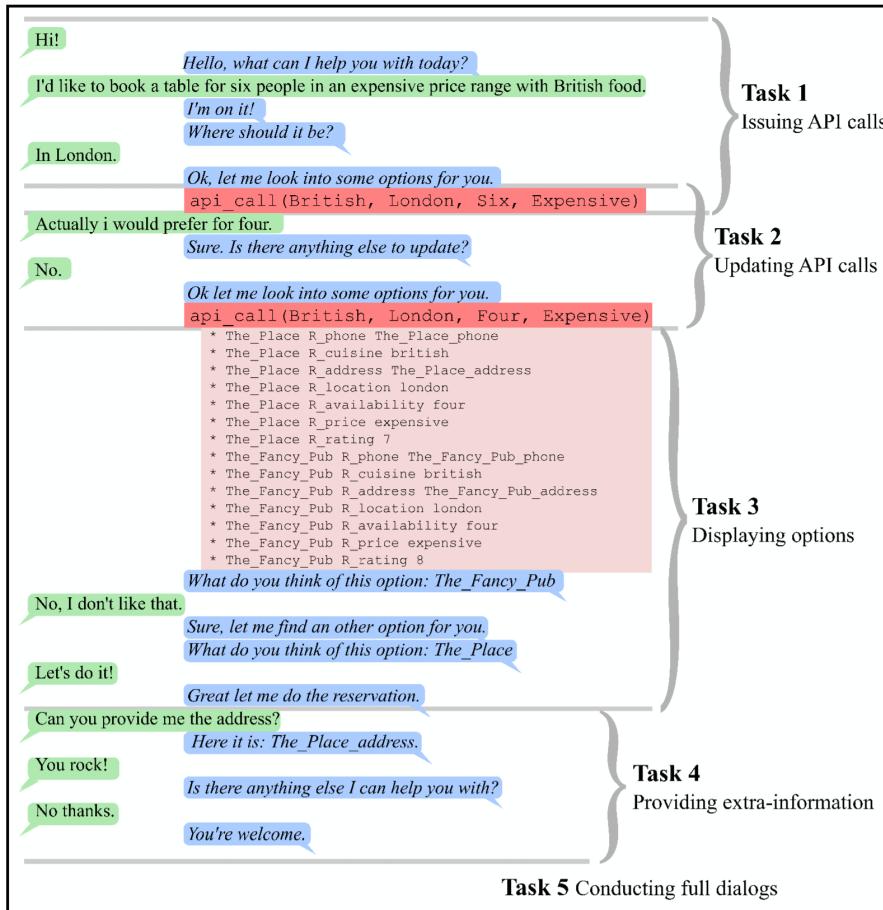


Figure 13: The 5 bAbi tasks for restaurant reservations. The dialog agent’s utterances are in blue, the user’s utterances are in green and API function calls are in red [Bordes et al., 2016].

A visualization of the different tasks is given in Figure 13. Various approaches to using KBs with neural models within the restaurant domain have been proposed [Wen et al., 2016, Eric and Manning, 2017, Williams et al., 2017]. Other bAbi tasks have also been explored with neural models augmented with KBs [Williams et al., 2017, Li et al., 2016b], for example dialog based reasoning [Weston et al., 2015] or movie recommendation tasks [Miller et al., 2016, Dodge et al., 2015]. A more general recommendation based dialog system is explored in [Yin et al., 2017]. The system presented in this work uses an information extraction component which takes in a query and returns results from a search engine.

A number of papers address the problem of integrating open-domain and task-specific approaches into one conversational model [Zhao et al., 2017a, Yu et al., 2017, Akasaki and Kaji, 2017]. Some of these have also been described in Section 3.2.4, since they usually make use of a knowledge base. Essentially, the issue is that the bAbi toy tasks mentioned above do not represent real-life interactions between users and task-oriented dialog systems. In order for conversational models to remain robust they have to be able to handle out of domain utterances. Otherwise users might get annoyed with the program, for example many rule-based systems might resort to output responses like *Sorry, I couldn't understand that, but I am happy to talk about movies.* in order to get the user back on track. One approach to handle these issues is to build a model that is able to differentiate between in-domain and non-task user utterances. Then, two different models trained separately on open-domain and task-specific datasets can produce response candidates and a scoring function can output the most probable response [Akasaki and Kaji, 2017].

### 3.3.3 Reinforcement Learning

Reinforcement learning (RL) [Sutton and Barto, 1998] is a type of learning framework which has seen increasing success recently [Mnih et al., 2013, Mnih et al., 2015, Silver et al., 2016]. RL performs very well in tasks where there isn't a defined loss function or it is not known what the gold truths are. Instead, learning is implemented using a reward which is automatically given to the model based on its state. A general visual description of the reinforcement learning framework can be seen in Figure 14.

In RL an agent is some kind of function consisting of parameters that are optimized with a learning algorithm. The agent receives as input the state of the environment and can output actions based on it. The goal is to maximize the expected reward through a good combination of actions. The environment is the task or setting in which the agent is situated. One of the most popular applications of RL is to games, since it is not known what the correct actions are in all steps of game [Mnih et al., 2013]. Instead, games have a clear end-goal and a learning agent can be rewarded based on whether it achieves the end-goal through its actions. An *episode* refers to a sequence of actions and states  $(s_1, a_1, \dots, s_t, a_t, s_T)$  until the agent reaches a terminal state  $s_T$ . The terminal state is referred to the state in which the agent receives a reward for the episode. This process, often called a markov decision process (MDP) consists of the triplet  $(S, A, R)$ , where  $S$  is the set of states,  $A$  is the set of actions and  $R$  is the reward for the episode [Kandasamy et al., 2017]. A policy  $\pi$  is a function that the agent implements in order to select an action to a given state. Stochastic policies  $\pi(a|s)$  represent the probability of an agent executing action  $a$  in state  $s$ .

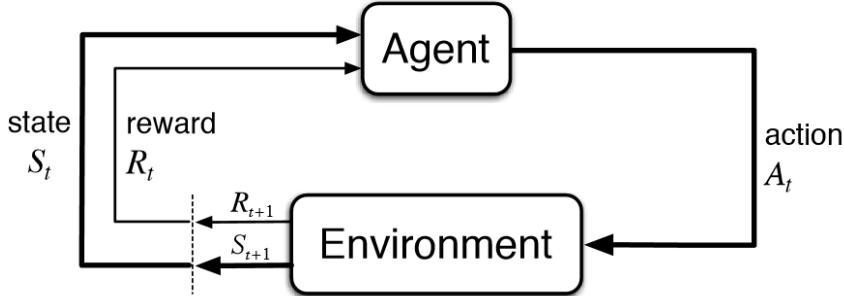


Figure 14: The reinforcement learning framework [Sutton and Barto, 1998].

The application of RL to neural conversational models is fairly straightforward. For a dialog agent the observed state at time-step  $t$  consists of the input sentence and what the agent has already outputted so far. The actions are the words in the vocabulary, since at any time-step it can only produce a word from the vocabulary. When the agent finishes generating the reply a reward is observed, which can be simply the difference between the generated sentence and the gold truth. However, other more sophisticated rewards are usually used, which are described in the next paragraph. Thus, the agent has to maximize this reward through a sequence of actions, generating one word at each time-step. The policy according to which the agent takes actions can be implemented as a normal seq2seq model. Generally, to train RL dialog agents the future reward of the episode has to be estimated at each time-step in order to get a reward for each action. This is usually done via the REINFORCE algorithm [Williams, 1992]. However, this approach is not perfect for dialog agents. Since rewards can only be observed at the end of an episode, all actions in that episode get the same reward. This means that if a response is of poor quality, like answering *I don't know* to the question *What's your name?* even the word *I*, which is mostly neutral will receive a negative reward. Instead as is customary in RL literature, a different reward for each action should be given. A possible solution is offered in [Kandasamy et al., 2017] using batch policy gradient methods. A different solution uses Monte Carlo search in order to sample tokens for partially decoded sentences [Li et al., 2017]. If there is a reward function, which is similar to a loss function, the agents can be trained by simple stochastic gradient descent.

Reward functions are often hand-crafted, meaning that the mathematical formulation of important properties of conversations is required in the beginning. In [Li et al., 2016c] the weighted sum of three different reward functions is used. The first one attempts to make responses easy to answer to. A set of dull and boring responses  $S$  is constructed, and if the response to the current action  $a$  is similar to these, then  $a$  is given a negative reward:

$$r_1 = -\frac{1}{N_S} \sum_{s \in S} \frac{1}{N_s} \log p_{seq2seq}(s|a) \quad (19)$$

where  $N_S$  denotes the cardinality of  $S$  and  $N_s$  denotes the number of tokens in  $s$ .  $p_{seq2seq}$  represents the probability output from a standard seq2seq model. The second reward attempts to capture the information flow in a conversation. More specifically, each utterance should contribute with new

information and it shouldn't be repetitive compared to previous ones. In order to achieve such a reward the semantic similarity between sentences can be penalized by

$$r_2 = -\log \frac{\mathbf{h}_{p_i} \mathbf{h}_{p_{i+1}}}{\|\mathbf{h}_{p_i}\| \|\mathbf{h}_{p_{i+1}}\|} \quad (20)$$

where  $\mathbf{h}_{p_i}$  and  $\mathbf{h}_{p_{i+1}}$  denote hidden representations from the encoder RNN for two consecutive turns  $p_i$  and  $p_{i+1}$ . The final reward function focuses on semantic coherence. This rewards generated responses that are coherent and grammatical by taking into account the mutual information between action  $a$  and previous dialog turns:

$$r_3 = \frac{1}{N_a} \log p_{seq2seq}(a|q_i, p_i) + \frac{1}{N_{q_i}} \log p_{seq2seq}^{backward}(q_i|a) \quad (21)$$

where  $p_{seq2seq}(a|q_i, p_i)$  denotes the probability computed by a seq2seq model of generating response  $a$  given previous dialog utterances  $[p_i, q_i]$ .  $p_{seq2seq}^{backward}$  is a seq2seq model trained with swapped targets and inputs as discussed in Section 3.1.2.  $N_a$  and  $N_{q_i}$  are the number of tokens in utterances  $a$  and  $q_i$  respectively.

Similarly in [Yu et al., 2017], a linear combination of four reward functions is proposed. In this work the functions implemented try to address turn-level appropriateness, conversational depth, information gain and conversation length in order to produce better quality responses. In [Yao et al., 2016] the reward function is based on calculating the sentence level inverse document frequency [Salton and Buckley, 1988] of the generated response.

However, hand-crafting reward functions is not ideal, since it might not be known exactly what conversational properties should be captured and how they should be formulated mathematically in an expressive way. A solution to this problem is to let agents figure out the appropriate reward by themselves, or use another agent to assign the reward. This has been explored before by using an adversarial approach [Li et al., 2017]. Generative adversarial networks [Goodfellow et al., 2014] consist of two networks competing with each other. A generator network (in this case a seq2seq model) tries to generate responses that mimic the training data and a discriminator network (implemented as an RNN based binary classifier in this case) tries to figure out whether the generated response came from the dataset or the generator network. Both of the networks are trained on whether the discriminator guessed correctly. Hence, the reward for the generator network comes from the discriminator network and represents whether the generator managed to fool the discriminator.

In [Havrylov and Titov, 2017] and [Kottur et al., 2017] a different line of research is explored using similar methods. The setting is still conversational, however conversation arises from a two-agent collaborative game, and it is argued whether natural language-like dialog can arise from such a setting. The sender agent (implemented as an RNN) receives as input an image and outputs a message, while the receiver agent (implemented as an RNN) receives as input this message and a set of pictures  $K$  and outputs a probability distribution over the pictures. The probability for a picture  $k \in K$  gives the likeliness of that picture being the one shown to the sender agent. Thus, the sender agent has to formulate a message which contains enough information so that the receiver

can choose between the pictures. This is a collaborative RL setting, since both agents receive the reward related to whether the receiver agent choose the right picture.

The approaches mentioned above are all off-line, since the reward is assigned by a predetermined reward function, and learning is based on a training dataset. However, on-line reinforcement learning has also been explored in a conversational setting [Li et al., 2016b]. In on-line learning the reward for generating a response is given by a human teacher interacting with the agent. The teacher can rate the response of the system on a scale and assign a numerical reward to it. This approach is very advantageous when a labeled dataset of dialogs is not available. The drawback is that human supervision is required.

### 3.3.4 Different Encoder-Decoder Models

So far the underlying architecture of encoder-decoder models was based on recurrent neural networks. However, attempts have been made towards using other types of neural networks as well. Recently, seq2seq models based solely on convolutional neural networks (CNNs) have achieved very good results in the field of NMT [Kaiser and Bengio, 2016, Kaiser et al., 2017a, Kalchbrenner et al., 2016, Gehring et al., 2017]. CNNs are a variant of feed-forward neural networks, that use a shared-weights architecture. They are the dominating architecture in the image processing domain [Krizhevsky et al., 2012]. CNN encoder-decoder models have not yet been applied to conversational modeling, but since the original RNN based seq2seq also originated from NMT, it is important to mention them. The appeal of CNN based models is that they are much faster to compute than RNNs, because they can be better parallelized, since they don't use recurrence.

A fully attention based encoder-decoder model was presented in [Vaswani et al., 2017], described in detail in Section 4.1. A more general encoder-decoder model was presented in [Kaiser et al., 2017b]. This model made use of all kinds of neural architectures like CNNs, attention mechanisms and sparsely-gated mixture-of-experts [Shazeer et al., 2017]. The generality and level of abstraction employed in the model makes it possible to achieve good results across various tasks and even modalities (audio, vision, text) by being trained simultaneously for all these tasks and by using the same parameters. A general diagram of the model can be seen in Figure 15.

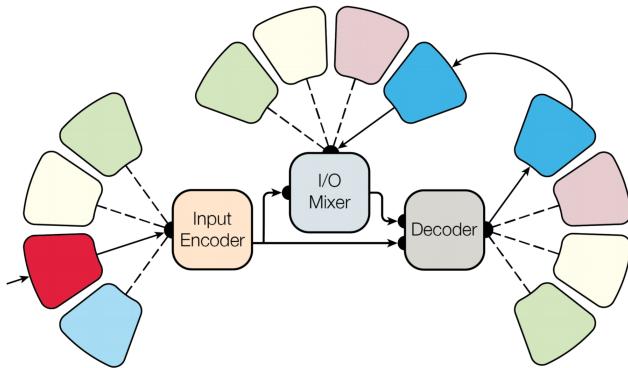


Figure 15: A high-level visualization of the MultiModel [Kaiser et al., 2017b]

Inputs coming from different modalities are encoded into common abstract representations using modality nets. The architecture of the modality net depends on the type of input. Additionally modality nets are also used to transform outputs into correct modality data. Between these modality nets a single encoder-decoder model transforms input representations to output representations. The decoder is autoregressive, meaning that it takes as input its previous output as well. The architecture of this model is very different from a standard RNN or CNN based seq2seq model, since it uses several types of building blocks like attention, mixture of experts and convolutional layers. It is not detailed here further, but the exact architecture can be found in Section 2 of the paper [Kaiser et al., 2017b].

## 3.4 Criticism

A conversation is a complex information exchanging process. In this section it is argued, based on recent literature, that current approaches to conversational modeling are not entirely suitable for the task. First, in Section 3.4.1 it is shown why some of the datasets used to train dialog agents are inherently not appropriate. Then, arguments are given regarding the reasons behind the standard loss function used to train conversational models failing to capture an important property of dialogs in Section 3.4.2. The memory of dialog agents is also an issue discussed in Section 3.4.3. Finally, in Section 3.4.4 it is shown why some of the standard evaluation metrics used to compare and validate various models perform poorly and do not provide meaningful comparisons.

### 3.4.1 Datasets

This work is mainly focused on open-domain dialog agents. For such agents a large quantity of data is necessary in order to be able to learn about various topics and properties of conversations. A good overview of datasets available for training dialog agents is presented in [Serban et al., 2015]. Large datasets are usually noisy, as is the case with the OpenSubtitles dataset [Lison and Tiedemann, 2016, opensubtitles.org, 2017], where the correct turn segmentation is not even known and usually it's assumed that alternate sentences are spoken by alternate characters [Vinyals and Le, 2015]. Furthermore, movie dialogs do not represent real, natural conversation since they are hand-crafted and often the conversations themselves are goal-oriented, involving outside factors related to the current scene in the movie. A good argument is given in [Danescu-Niculescu-Mizil and Lee, 2011]: *"For example, mundane phenomena such as stuttering and word repetitions are generally nonexistent on the big screen. Moreover, writers have many goals to accomplish, including the need to advance the plot, reveal character, make jokes as funny as possible, and so on, all incurring a cognitive load."*.

The other prevalent type of dataset is based on messageboards and post-reply websites, like the Ubuntu dialog corpus [Lowe et al., 2015] or Twitter [Shang et al., 2015, Li et al., 2016a, Jena et al., 2017]. The underlying problem with these datasets is that they don't actually have one on one conversations. Rather there is usually a post to which a multitude of people can reply, thus multi-turn dialogs are rare. Secondly, the conversations are public, which in my opinion are not as natural as private ones. Since the goal is to build an open-domain chatbot that can hold private

conversations, the data that it was trained on should also be similar in nature.

### 3.4.2 The Loss Function

In seq2seq models the standard loss function is based on calculating how different the predicted response probabilities and the gold truths are. This function was borrowed from NMT, where it performs better, since for each input sentence there is usually a somewhat less ambiguous output translation. However, even for NMT it does not perform perfectly, since a sentence can be formulated in a number of ways. Consequently for conversational modeling it's even worse, since a source utterance can have a multitude of suitable replies, which can be very different from each other semantically. For example the reply to *How was your day?* can be as simple as *Okay.* or as long as a whole paragraph describing events that happened during the day. Since the task of conversational modeling is so ambiguous the standard loss function is not suited for training good chatbots [Vinyals and Le, 2015, Li et al., 2015]. In many works it has been shown that dialog agents tend to output generic responses like *I don't know* [Vinyals and Le, 2015, Serban et al., 2016, Li et al., 2015, Li et al., 2016a, Jena et al., 2017]. My presumption is that it is precisely because of the loss function that the models learn to output safe and generic responses. Since the task of conversational modeling is ambiguous and even in a training set there are various replies to the same utterance, the loss function makes the model learn an average of these responses. Utterances can be represented by the word embeddings that make them up, many dimensional real-valued representations. Hence, if there are multiple semantically different replies to the same source utterance, the model will learn to average these out because of the loss function. Furthermore, in the vector space this average will point close to where generic and safe answers are located, since these have been seen by the model as replies for various source utterances and they don't carry much information. Thus, the model learns to output these neutral responses, because it was trained with a loss function that tries to average out ambiguity.

Attempts have been made to address these issues and to define better loss functions, presented in Section 3.1.2. New ideas regarding the loss function issue are proposed in Section 6.1.

### 3.4.3 Memory

A more specific problem with current conversational models is that they don't have any memory. While architectures have been devised in order to take into account previous turns, they are still limited. For example they can't take into account hundreds of previous turns. A user would expect from a chatbot that once he/she talks about something, the chatbot will mostly remember the dialog, as is normal in human-human conversations. The chatbot should be able to learn new things about the user in order to make the experience personalized. Currently there aren't any attempts to address this issue, since it is not conversational, but rather it arises from the deployment of dialog systems to the real world. A novel idea that could solve this issue is proposed in Section 6.2.

### 3.4.4 Evaluation Metrics

It has been shown that standard metrics like perplexity and Bleu, don't correlate at all with human judgment for evaluation of dialog agents [Tao et al., 2017, Liu et al., 2016]. Figure 16 depicts the correlation of Bleu with human judgment for evaluating a seq2seq model trained on two different datasets. I believe that the reason for this is similar to the loss function issue. Namely, these metrics assume that there is only one gold truth response for each source utterance, because they are based on measuring the difference between the predicted and the gold truth response. Since this is not the case, they will assign low scores to responses that were deemed natural by human judges, but aren't similar to the gold truth responses.

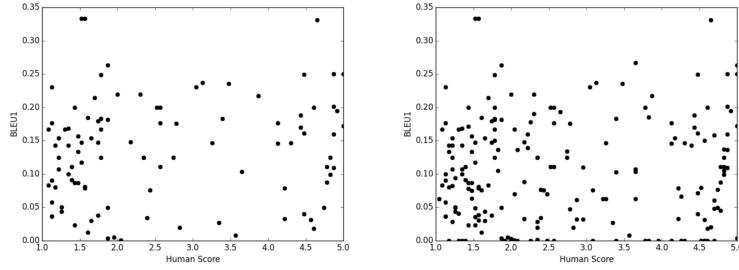


Figure 16: Bleu score correlation with human scores [Liu et al., 2016] on a Twitter corpus (left) and on the Ubuntu dialog corpus [Lowe et al., 2015] (right).

Furthermore, while human evaluation is one of the most accurate metrics it still has its drawbacks. Most importantly, it is not automatic and thus can be costly and can not be standardized.

Attempts have been made in order to design better automatic evaluation metrics based on neural networks [Tao et al., 2017, Lowe et al., 2017, Li et al., 2017]. Since they are neural network based they can be trained with labeled data. They receive a source utterance and the reply to it as input and they have to output a score. Then, this score can be compared with the true score assigned by a human annotator and thus the model learns from its errors. As can be seen in Figure 17 these metrics indeed correlate better with human scoring.

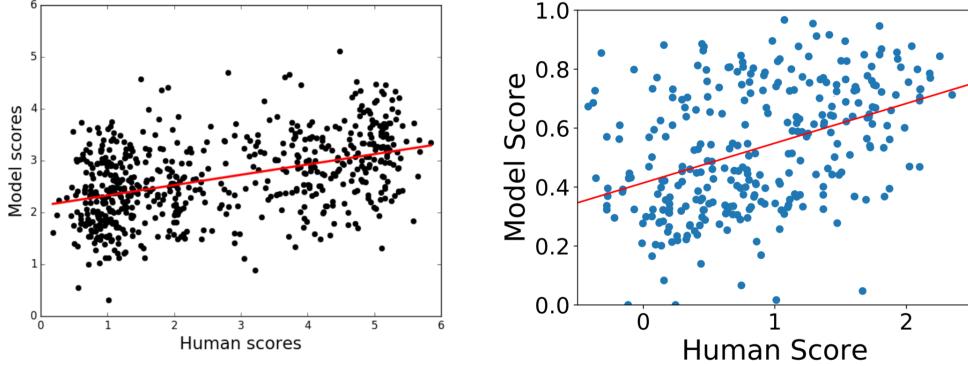


Figure 17: Correlation of ADEM (left) [Lowe et al., 2017] and RUBER (right) [Tao et al., 2017] with human scorers.

### 3.5 Summary

In this section an in-depth survey of recent literature related to conversational agents has been presented. Additional, but essential details of encoder-decoder models have been described, like the context of conversations, objective functions and evaluation methods in Section 3.1. Then, in Section 3.2 an overview of various techniques used to augment the performance of seq2seq models was given, like attention, pretraining, additional input features, knowledge bases and copying. The literature survey was finished by describing different encoder-decoder architectures in Section 3.3, like hierarchical models, task-oriented systems, reinforcement learning based agents and other (non-RNN based) encoder-decoder models.

Finally, in Section 3.4 it was argued that some of the current approaches to build conversational models are inherently unsuitable for the task. Arguments were presented related to current datasets not providing natural conversational data and to how the standard loss function used to train conversational models fails to capture important properties of dialogs. The criticism was finished by showing that current dialog agents lack memory, an essential part of conversations and why standard evaluation metrics are unsuitable to evaluate dialog agents.

## 4 Experiments

In this section preliminary experiments with the Transformer encoder-decoder model are presented, which is described in detail in Section 4.1. The model is trained on conversational data, which hasn't been done previously, since the architecture is very recent and it was originally created for NMT [Vaswani et al., 2017]. It is important to mention, that these experiments are just a first step, and they are not meant to provide a complete and thorough analysis of the transformer model applied to conversational data. In Section 4.2 a brief overview of the datasets used for training is given. Finally, in Section 4.3 a detailed description of the training setups that were run is given.

### 4.1 The Transformer Model

The Tranformer is an encoder-decoder model based solely on attention mechanisms and feed-forward neural networks, achieving state-of-the-art results in NMT tasks [Vaswani et al., 2017]. In this section its architecture is described in detail, which can be seen in Figure 18.

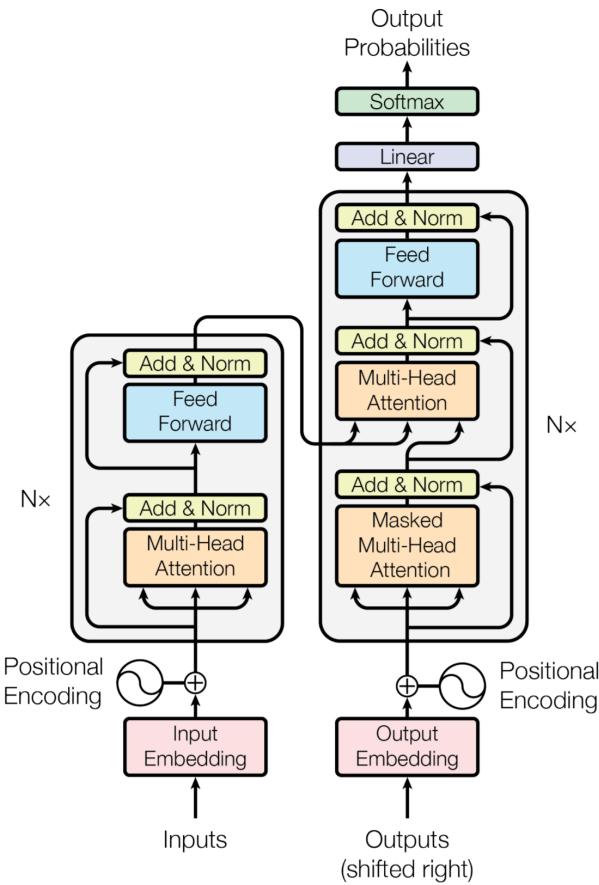


Figure 18: The architecture of the Transformer model [Vaswani et al., 2017]. The encoder network can be seen on the left and the decoder network on the right.

The model is made up of two sub-networks, the encoder and the decoder networks. The encoder network takes as input the word embeddings of the source sentence and produces a representation. Then, the decoder network takes as input this representation and the word embeddings of already generated tokens from the output sentence. Based on these it produces output probabilities for one word in the sentence at a time. The decoder is auto-regressive, because at each step it generates predictions based on previously predicted words. First the high-level architecture of the encoder and the decoder networks is presented in Section 4.1.1. Then a detailed description of the building blocks used is given, like attention (Section 4.1.2) and feed-forward networks (Section 4.1.3). Finally, further methods used are presented, like positional encoding in Section 4.1.4 and regularization techniques in Section 4.1.5.

### 4.1.1 Encoder and Decoder Networks

The encoder network contains 6 identical layers. The output of a previous layer serves as input to the next layer. Within 1 layer, there are two main blocks. First, the input embeddings go through a multi-head self-attention block, described in Section 4.1.2. Then the outputs from this block serve as input to a position-wise fully connected network, described in Section 4.1.3. Both blocks are augmented with residual connections and layer normalization, presented in Section 4.1.5.

Similarly, the decoder network also consists of 6 layers. However, it contains 3 blocks, the first and third basically being the same as the two blocks in the encoder network. The additional middle block is a multi-head attention mechanism over the output of the encoder, connecting the two networks. An important difference in the decoder’s first multi-head self-attention block is that it takes as input already generated word embeddings from the output sentence. Since the size of the inputs is fixed, not yet generated output embeddings are masked in order to ensure that they don’t take part in computing the attention. This, together with shifting the output embeddings by one position ensures that predictions for position  $i$  in the output sentence only depend on previously generated words at positions less than  $i$ .

### 4.1.2 Attention Mechanisms

The attention mechanism is implemented as multi-head scaled dot-product attention over query, key and value vectors ( $q, k, v$ ). In self-attention blocks, these vectors all come from input embeddings. Each input vector is projected three times with separate projections to get queries, keys and values of same dimensions as the original input. In the attention block which connects the encoder and decoder networks the queries are the outputs of the previous attention block in the decoder and the key and value vectors come from projecting the output of the final encoder layer two times with separate projections. In the first self-attention block in the decoder, masking is used by setting all values to  $-\infty$  in the input of the softmax functions.

For computing the scaled dot-product attention, which is also described in Section 3.2.1, the following equation is used:

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^\top}{\sqrt{d_k}} \right) V \quad (22)$$

where  $Q, K, V$  are matrices built from the query, key and value vectors respectively across the whole input sequence.  $d_k$  is the dimension of the key vectors.

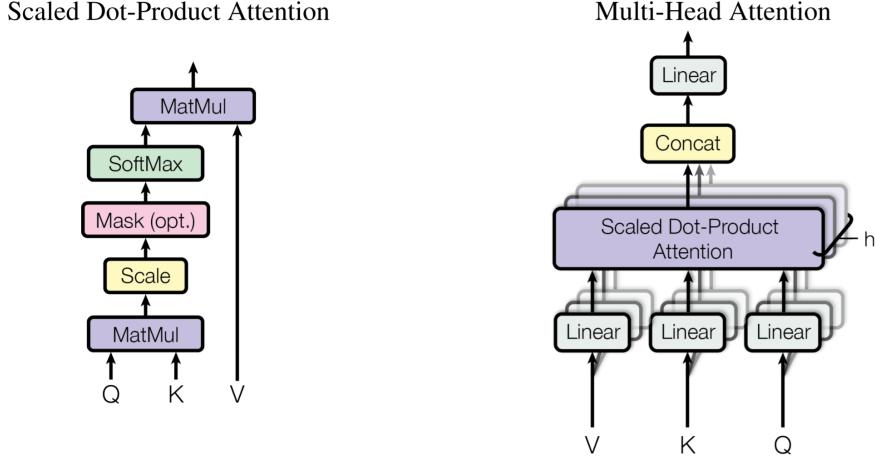


Figure 19: A diagram of the scaled dot-product attention can be seen on the left, and of the multi-head attention on the right [Vaswani et al., 2017].  $Q, K, V$  are the query, key and value matrices respectively and  $h$  denotes the number of attention heads.

Multi-head attention (Figure 19) means that there are several parallel channels of attention computations over separate inputs. To get separate inputs, the query, key and value matrices are linearly projected to a dimension size that is equal to their original dimension divided by the number of heads. For each head and each vector the projection is done via a separate learned weight matrix. This way instead of computing the attention over the whole  $d_{model} = 512$  dimensional inputs at once, the computation is divided into  $h = 8$  heads over  $\frac{d_{model}}{h} = 64$  dimensional query, key and value vectors (Equation 23). Finally, the outputs of all heads are concatenated and once again projected, resulting in the final output value of the attention block (Equation 24).

$$\text{head}_i = \text{Attention}(QW_{Qi}, KW_{Ki}, VW_{Vi}) \quad (23)$$

$$\text{MultiHead}(Q, K, V) = \text{Concatenate}(\text{head}_1, \dots, \text{head}_h)W_O \quad (24)$$

where  $W_{Qi}, W_{Ki}, W_{Vi}, W_O$  are learned weight matrices.

#### 4.1.3 Feed-Forward Networks

The feed-forward (FFN) blocks are implemented as two consecutive convolutions with kernel size 1, with a ReLU activation between them (Equation 25). This means that each filter contains a singular weight, and this weight is multiplied with each element from the input vector separately to produce an output vector:

$$\text{FFN}(\mathbf{x}) = \max(0, \mathbf{x}W_1 + \mathbf{b}_1)W_2 + \mathbf{b}_2 \quad (25)$$

where  $\mathbf{x}$  is the input vector,  $W_1$  and  $W_2$  are the weight matrices and  $\mathbf{b}_1$  and  $\mathbf{b}_2$  are the bias vectors.

#### 4.1.4 Positional Encoding

Since the model doesn't use convolutions or recurrence it doesn't have any positional information about the words in a sequence. In order to address the issue, positional encodings are used. Each word has a separate representation related to its position in the sentence. They are the same size as the word embeddings, so the two vectors can be summed up, to produce a final representation for each word. The positional encoding vectors are not learned, but rather constructed using sine and cosine functions of different frequencies:

$$PE[pos_j, 2i] = \sin\left(\frac{pos_j}{10000^{\frac{2i}{d_{model}}}}\right) \quad (26)$$

$$PE[pos_j, 2i + 1] = \cos\left(\frac{pos_j}{10000^{\frac{2i}{d_{model}}}}\right) \quad (27)$$

where  $PE[pos_j]$  is the positional encoding for the  $j$ -th word in the sequence and  $i \in \{0, 1, \dots, d_{model} - 1\}$ .

#### 4.1.5 Regularization and Other Techniques

Layer normalization [Ba et al., 2016] is used over each block of the model. This is a regularization technique which computes statistics over a layer and normalizes the output of the layer based on them. Additionally, layer normalization is implemented over the sum of the output of the layer  $Layer$  and the original input  $X$  to that layer:

$$\text{LayerNormalization}(X + \text{Layer}(X)) \quad (28)$$

This connection between the input and output of a layer is called a residual connection [He et al., 2016].

Furthermore, dropout [Srivastava et al., 2014] is employed as a regularization technique. Dropout is a stochastic function that takes as input an array and replaces each element with 0 according to a given probability. Dropout is applied to the output of each block (before layer normalization) and during the computation of the scaled dot-product attention.

## 4.2 Datasets

In this section the two datasets that were used for training conversational models are presented. Since the Transformer model has been adapted without any modifications from NMT, all data consists of source-target utterance pairs. First, in Section 4.2.1, an overview of the Cornell Movie-Dialog Corpus [Danescu-Niculescu-Mizil and Lee, 2011] and of the preprocessing techniques applied is given. This overview is also given similarly for the OpenSubtitles Corpus [Tiedemann, 2009] in Section 4.2.2.

### 4.2.1 Cornell Movie-Dialog Corpus

This corpus contains 220579 conversational exchanges between 10292 pairs of movie characters extracted from 617 movies. Dialogs are turn segmented, meaning that one utterance can contain multiple sentences. There's also metadata included, for example for each utterance the character's name that utters it. About 200K utterances were used as training data and 20K utterances as validation data. It is important to mention that if an utterance is not the first and not the last in a conversational exchange then it will be present in both the source and target data. For example a conversation consisting of 3 turns is split into 2 source-target pairs. The first pair consists of the first utterance as source and the second as target and the second pair consists of the second utterance as source and the third as target. This means that the majority of the utterances are present both in the source data and the target data.

For preprocessing all the characters from the corpus that weren't in the set ( $a - z.?!'$ ) were simply deleted. Furthermore, all words that contained the ' symbol were split into two according to this rule:  $I'll=I 'll$ . Words containing the sequence of characters  $n't$  were not subjected to the previous rule, rather they were split into two according to this rule:  $don't=do n't$ . This is the standard method for English tokenization.

### 4.2.2 OpenSubtitles Corpus

The OpenSubtitles Corpus contains movie subtitles made by [opensubtitles.org, 2017]. Unfortunately, this corpus only contains sentence-level segmentation, which makes it noisy, since it might be the case that an utterance contains multiple sentences. Furthermore, it is extracted from subtitles, that contain not only dialogs but also scene descriptions and other non-conversational sentences. In this work the 2016 version of the corpus is used [Lison and Tiedemann, 2016], but only a subset consisting of 62M sentences for training and 28M sentences for validation is kept. The reason behind using only a subset of the corpus is that it makes the dataset similar in size to [Vinyals and Le, 2015].

Following [Vinyals and Le, 2015] each alternate sentence is considered as an utterance spoken by a different character. Similar to the Cornell-Movie Dialog data, all the sentences appear in both the source and target data. Preprocessing was mostly done in the same fashion as well.

## 4.3 Training Details

In this section various training setups are described. In Section 4.3.1 the Transformer implementation is described and how it was integrated with the novel datasets. Then, from Section 4.3.2 to Section 4.3.5 the types of data and hyperparameter values used for 4 different trainings are described. For all trainings the vocabulary is constructed based on most frequent words in the training data.

### 4.3.1 Tensor2Tensor

The official implementation of the Transformer model in Tensorflow<sup>1</sup> was used. This implementation is part of a bigger framework, the Tensor2Tensor library<sup>2</sup>. In short, to run a training using this library, a model, a problem and a hyperparameter set needs to be defined. In this work the Transformer model from the library was used without any modifications and I defined my own problems and hyperparameter sets. This is basically done by sub-classing already existing classes, implementing own functions and then registering the classes so that the library sees them and they can be used the same way as the already defined classes. The whole code that was written to implement the problems and hyperparameter sets can be found on Github<sup>3</sup>.

To implement my own data handling functions the *Text2TextProblem* class had to be sub-classed. With the preprocessing steps described in Section 4.2 1-1 source and target file was created for training and 1-1 source and target file was created for validation. Each line in the source file contains an utterance and the corresponding line in the target file contains the response to that utterance. These files were then given as inputs to the library, which created specially encoded and sharded files as the final dataset, ready to be used for training.

Similarly for the hyperparameters the *basic\_params1* class was sub-classed and I defined my own hyperparameter values. The specifics for each training setup can be found in the following sections.

### 4.3.2 Cornell Movie Training

For this training the Cornell Movie-Dialog dataset was used as described in Section 4.2.1. The most frequent 32765 words in the training data were kept as vocabulary. Additionally the *<pad>* token was used for padding input sequences to same lengths, the *<EOS>* token was used to signal the end of an utterance and the *<UNK>* token was used to replace all words not present in the vocabulary. In total this gives a vocabulary size of 32768.

For hyperparameters the same ones were used as in [Vaswani et al., 2017] for the base variant of the Transformer model. Because of memory issues the batch size was limited to 4096 tokens/batch. With this batch size the model was trained for a total of 350k steps, which took about 2.5 days on a Nvidia GeForce GTX 1070 GPU.

### 4.3.3 Cornell Movie Training with Speakers

For this training the data used in the previous section was augmented with speaker-addressee character labels, similar to [Li et al., 2016a]. The dataset was constructed by starting each line in the source data with a special token that denotes the character that utters that line and ending it with a special token denoting the character to which the utterance is addressed. The target data does not contain any character tokens. Since characters are specific to movies, it is important to mention that characters with the same name, but from different movies were represented by different

---

<sup>1</sup><https://github.com/tensorflow/tensorflow>

<sup>2</sup><https://github.com/tensorflow/tensor2tensor>

<sup>3</sup><https://github.com/ricsinaruto/Seq2seqChatbots>

tokens. Because of this the training and validation sets were constructed by splitting dialogs. If the two sets would have been constructed via splitting by movies, then all the character names in the validation set would not have been seen during training. This is unwanted, since the goal is for the model to learn something about the various characters' dialog styles and personalities. The 31996 most frequent words were used for vocabulary. Additionally, the 8000 most frequent character names (separated by movies) were added to the vocabulary and all character names less frequent were replaced with the *<UNK NAME>* token. Together with the 3 special tokens mentioned in the previous section this results in a total vocabulary size of 40K.

The same hyperparameters were used as in [Vaswani et al., 2017] for the base variant of the Transformer model. Because of memory issues the batch size was limited to 4096 tokens/batch. With this batch size the model was trained for a total of 238K steps, which took about 1.5 days on a Nvidia GeForce GTX 1070 GPU.

#### 4.3.4 OpenSubtitles Training

For this training the OpenSubtitles dataset, described in Section 4.2.2, was used. Together with the 3 special tokens mentioned in Section 4.3.2 the total vocabulary size is 100K, using the most frequent words in the training set.

For hyperparameters the same ones were used as in [Vaswani et al., 2017] for the base variant of the Transformer model. Because of memory issues the batch size was limited to 2048 tokens/batch. With this batch size the model was trained for a total of 1M steps, which took about 5.5 days on a Nvidia GeForce GTX 1070 GPU.

#### 4.3.5 OpenSubtitles Training Finetuned with Cornell Movie Data

This training combines the two datasets by first training on the OpenSubtitles dataset and then finetuning the model on the Cornell Movie-Dialog dataset with speaker embeddings.

The parameters of the model were initialized with the trained model described in Section 4.3.4. The model was further trained for 675K steps on the same data as in Section 4.3.3, which took about 3.5 days on a Nvidia GeForce GTX 1070 GPU. For vocabulary the 100K most frequent words from the initial OpenSubtitles training were kept and the 3000 most frequent character name tokens (separated by movies) from the Cornell Movie-Dialog dataset were added, resulting in a total vocabulary size of 103K. For hyperparameters the same ones were used as in [Vaswani et al., 2017] for the base variant of the Transformer model. Because of memory issues the batch size was limited to 2048 tokens/batch.

## 5 Results

In this section a detailed analysis of the trained models is given. They are also compared with a baseline seq2seq model presented in [Vinyals and Le, 2015], which was trained on OpenSubtitles data. In Section 5.1 a quantitative analysis is given by computing standard metrics. Then, in Section 5.2 a qualitative analysis is given by comparing sample output responses.

### 5.1 Quantitative Analysis

Metrics	S2S	Cornell	Cornell S	OpenSubtitles	OpenSubtitles F
Perplexity	17	17	15.4	11.7	24.6
Bleu	-	4.7	4.2	6.8	4.6

Table 1: Perplexity and Bleu scores on validation data. S2S stands for the baseline seq2seq model, Cornell stands for the training setup presented in Section 4.3.2, Cornell S stands for Section 4.3.3, OpenSubtitles stands for Section 4.3.4 and OpenSubtitles F stands for Section 4.3.5.

In Table 1 the perplexity and Bleu scores of the trained models can be seen on the respective validation datasets. Since the models trained on Cornell Movie-Dialog data overfitted quite quickly the dataset as can be seen in Figure 20 and Figure 21, the perplexity and Bleu scores presented are computed on the validation data before overfitting occurred. This means that for the Cornell model the scores were computed after 20K training steps and for the Cornell S model, scores were computed after 16K steps. For the OpenSubtitles training the model didn't seem to overfit, however after a quick drop the loss stagnated for the remainder of the training, which could be because the learning rate was not adequate or the model was too small to be able to learn more. Future work should address these issues, described in detail in Section 6. The effects can be seen in Figure 22 up to step 1M, since afterwards the training was switched to the Cornell Movie-Dialog dataset. The scores corresponding to the OpenSubtitles F training are very poor, since the model immediately started to overfit after being switched from OpenSubtitles to Cornell Movie-Dialog data starting from step 1M in Figure 22.

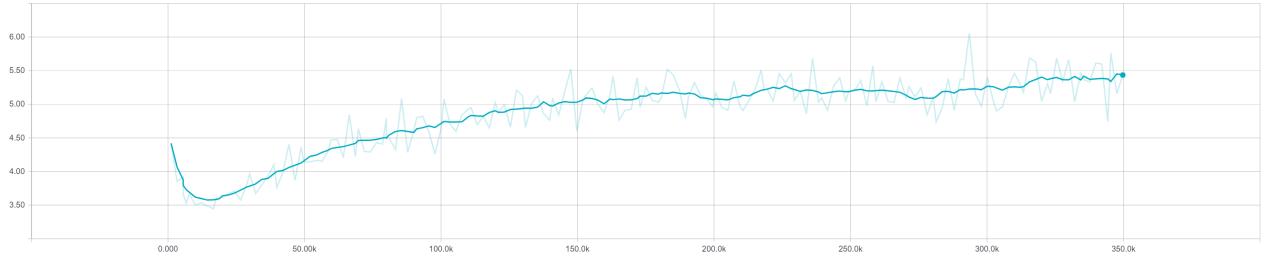


Figure 20: The diagram shows the validation loss over the entire training of the Cornell model. The vertical axis represents the loss value and the horizontal axis represents the number of training steps.

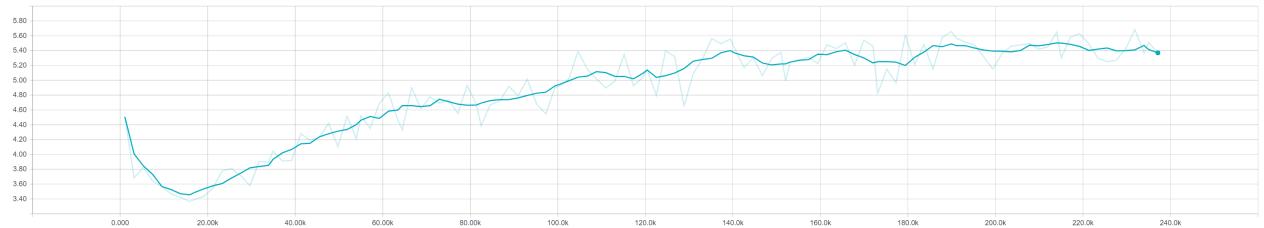


Figure 21: The diagram shows the validation loss over the entire training of the Cornell S model. The vertical axis represents the loss value and the horizontal axis represents the number of training steps.

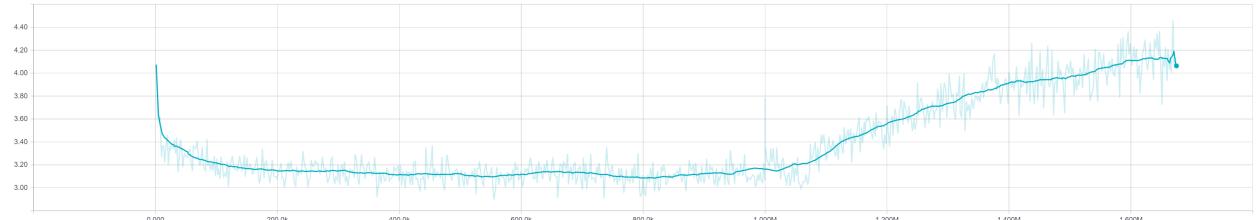


Figure 22: The diagram shows the validation loss over the entire training of the OpenSubtitles and the OpenSubtitles F model. In the first 1M steps the model was trained on OpenSubtitles data, then it was switched to speaker-annotated Cornell Movie-Dialog data. The vertical axis represents the loss value and the horizontal axis represents the number of training steps.

## 5.2 Qualitative Analysis

In this section output response samples from the various trainings are presented. For the sake of comparison with the baseline seq2seq model, the source utterances are a subset of the ones used in [Vinyals and Le, 2015]. The source utterances were divided into several categories like in [Vinyals and Le, 2015]. Generated responses by the trained models to the various source utterances can be seen from Table 2 to Table 6. Interestingly the best responses from the two Cornell trainings were not those outputted before the models started to overfit. Before overfitting the models usually outputted very generic responses similar to the ones seen in the OpenSubtitles column of the tables. By letting the models overfit a little on the training data more diverse and interesting responses were generated. This further suggests that the loss function is not in direct correlation with the quality of responses, since it gives a better score to safe and generic answers, like *I don't know*. Specifically, the examples presented for the Cornell model are after training for 219K steps and for the Cornell S model after training for 238K steps. Also, for the OpenSubtitles F model, generated output responses after 370K steps of finetuning were used, since there was no further improvement observed in the quality of responses as training continued.

As can be seen in the tables, the OpenSubtitles F model, which was finetuned on Cornell Movie-Dialog data generates better responses than the OpenSubtitles variant, which proves that the finetuning had a beneficial effect on the quality of generated responses. This is again in contrast with the loss and metrics values, since these are much better for the OpenSubtitles model. Generally however the models trained solely on the Cornell Movie-Dialog dataset generate the most diverse and interesting responses. Despite this, none of the trainings achieved the performance of the baseline seq2seq model. Perhaps the most striking result is the OpenSubtitles model, since this provides the most direct comparison with the baseline seq2seq model, because similar and similarly sized datasets were used for training the two models. Several reasons are given for these results in Section 6, as well as possible solutions and future directions. It is important to note however that the source utterances were originally selected for evaluating the seq2seq model and they don't provide a complete qualitative picture of the performance of the conversational models, since for other source utterances the Transformer model based trainings might generate better results. In order to truly compare the trained models a set of random source utterances should be constructed and human evaluators employed to rank the generated responses.

Source Utterance	S2S	Cornell	Cornell S	OpenSub-titles	OpenSub-titles F
hello !	hello !	hello .	hi .	hello .	hi .
how are you ?	i 'm good .	fine .	fine . how are you ?	how are you ?	fine .
what 's your name ?	i 'm julia .	my name is sir robin of camelot !	david .	<UNK>.	bob .
when were you born ?	july 20th .	what ?	yesterday .	two years ago .	just a few minutes ago .
what year were you born ?	1977 .	eight years ..	my father was the paper .	i don 't know .	<UNK>.
where are you from ?	i 'm out in the boonies.	south america .	new york .	where are you ?	in here .
are you a man or a woman ?	i 'm a woman .	i 'm a teacher .	i never forget a pussy . . . cat .	yes .	yes .
why are we here ?	i 'm not sure .	we ca n't stay here .	the list .	i don 't know .	we 've got to get out of here .
okay bye !	bye .	bye .	bye !	bye .	bye .
see you later .	bye .	where you off to ?	bye !	bye .	bye .

Table 2: A set of general source utterances and the generated responses by the trained models.

Source Utterance	S2S	Cornell	Cornell S	OpenSub-titles	OpenSub-titles F
what do you like to talk about ?	nothing .	about things . about your wife .	about things that are important to me . . .	i don 't know .	i do <UNK>know .
what is your job ?	i 'm a lawyer .	my job my rules .	i need a vacation .	i don 't know .	i . . . i . . .
what do you do ?	i 'm a doctor .	i 'm a comedian .	i 'm taking a piss .	i don 't know .	i 'm sorry .

Table 3: Job and Personality.

<b>Source Utterance</b>	<b>S2S</b>	<b>Cornell</b>	<b>Cornell S</b>	<b>OpenSub-titles</b>	<b>OpenSub-titles F</b>
who is skywalker ?	he is a hero .	a snake !	ben .	i don 't know .	my brother .
who is bill clinton ?	he 's a billionaire .	name your brother .	i do n't know .	i don 't know .	he 's my brother .
is sky blue or black ?	blue .	blue eyes .	a car accident .	yes .	yes .
does a cat have a tail ?	yes .	does it matter ?	maybe .	no .	yes .
does a cat have a wing ?	no	it 's a <UNK>.	not for lack of trying .	yes .	yes .
can a cat fly ?	no .	you 're a woman are n't you ?	or a little bit .	no .	sure .
how many legs does a cat have ?	four , i think .	one .	four .	i don 't know .	four .
how many legs does a spider have ?	three , i think .	two .	thirty .	i don 't know .	four .
what is the color of the sky ?	blue .	blue !	green .	i don 't know .	blue .
what is the color of water ?	water .	what ?	white .	i don 't know .	blue .
what is the color of blood ?	it is the same as a black eye .	it is the rabbit .	beautiful .	i don 't know .	blue .
how much is two plus two ?	four .	i do n't know .	well do n't worry about it .	.	ten .
how much is ten minus two ?	seventy - two .	between ten and twelve .	yeah .	.	ten .

Table 4: General Knowledge Q&A.

<b>Source Utterance</b>	<b>S2S</b>	<b>Cornell</b>	<b>Cornell S</b>	<b>OpenSub-titles</b>	<b>OpenSub-titles F</b>
my name is david . what is my name ?	david .	victor .	<UNK>.	what 's your name ?	bob .
my name is john . what is my name ?	john .	i want to know .	what ?	<UNK>.	bob .

Table 5: How different models handle context.

<b>Source Utterance</b>	<b>S2S</b>	<b>Cornell</b>	<b>Cornell S</b>	<b>OpenSub-titles</b>	<b>OpenSub-titles F</b>
what is the purpose of life ?	to serve the greater good .	i do n't know .	my father .	i don 't know .	i do <UNK>know .
what is the purpose of living ?	to live forever .	i do n't know .	a lot of people .	i don 't know .	i do <UNK>know .
what is the purpose of existence ?	to find out what happens when we get to the planet earth .	i do n't know .	i do n't know . could be a lot of lives .	i don 't know .	i do <UNK>know .
where are you now ?	i 'm in the middle of nowhere .	here .	here .	where are you ?	in here .
what is the purpose of dying ?	to have a life .	i do n't know .	we 're not going somewhere .	i don 't know .	i do <UNK>know .
what is the purpose of being intelligent ?	to find out what it is .	intelligent beyond our comprehension .	we 're not sure .	i don 't know .	i do <UNK>know .
what is the purpose of emotions ?	i don 't know .	i do n't know sir .	you know a little of this a little of that .	i don 't know .	i do <UNK>know .

Table 6: Philosophical Q&A.

In Table 7 generated output responses from the Cornell S training (after 145K steps) to various source utterances can be seen. The text of the source utterances is the same for all columns, however the speaker and addressee names are changed. More specifically for the *Cornell S* column no names were given. For the columns *Cornell S matching* and *Cornell S matching R* the characters Ben and Mrs. Robinson were used from the movie *The Graduate*. These characters were selected because they have many interactions in the training data, thus the model must have learned something about them. The difference between the two columns is that in the *Cornell S matching* column Mrs. Robinson addresses Ben and in the *Cornell S matching R* column Ben addresses Mrs. Robinson. For the final two columns the character of Ben is kept, however the other speaker embedding used is that of Joe from the movie *Innerspace*. In the *Cornell S different* column Ben addresses Joe and in the *Cornell S different R* column Joe addresses Ben. These two characters were selected, because they have many utterances in the training data, however since they aren't in the same movie, they don't interact with each other.

There is a clear difference between the generated responses, solely in consequence of the speakers and addressees being different. The most striking influence can be seen where the source utterance asks the name of someone. For these questions the model learns to respond with actual corresponding names, however they are sometimes reversed. In the second row, when Mrs. Robinson asks *what's your name?* from Ben, the model responds with Mrs. Robinson and when Ben asks the question from Mrs. Robinson the model responds with Benjamin. In row 8, if the first sentence is ignored, the responses to the question *what is my name?* are the same as for the previous question, but since this question refers to the speaker, the replies are actually correct. It is important to note here that the tokens used for speakers and addressees are completely different from any names present in the dataset. Thus the model learned to come up with a sequence of 3 tokens (*mrs . robinson*) as answer, while the actual token used to represent Mrs. Robinson is *MRS.\_ROBINSON\_m77*.

Even when the characters did not interact in the training dataset, meaningful responses are generated. For example, in row 6 and column 5, when Ben says *see you later.* to Joe, the model responds with *benjamin ?*, which makes sense. Interestingly this was the only speaker-addressee combination where the model managed to answer with numbers to the utterances in the last two rows of the table. Moreover, in rows 8 and 9 for the question *what is my name?*, when Joe asks Ben, the model responds with *mrs . robinson*, which makes sense since the two characters haven't interacted before, thus if Ben is addressed it responds with the character name that it had the most interactions with.

The OpenSubtitles F training, which was also speaker annotated didn't produce such distinct and interesting answers, so the comparison of different source utterances is not presented here. This is probably due to the model being pretrained on the OpenSubtitles dataset. Since this dataset is much larger the model didn't manage to adapt to the smaller Cornell Movie-Dialog dataset, and thus its responses remained somewhat general and safe.

Source Utterance	Cornell S	Cornell S matching	Cornell S matching R	Cornell S different	Cornell S different R
how are you ?	fine . how are you ?	very well . thank you .	i 'm in touch .	i ca n't how are you ?	i 'm fine . what 's up ?
what 's your name ?	david .	mrs . robinson .	benjamin .	do n't you know ?	why do you do that ?
when were you born ?	yesterday .	very interesting .	about three years .	about a year ago .	tuesday .
where are you from ?	new york .	i 'm at the airport .	at the airport .	off the street .	with you .
why are we here ?	the list .	well i just want you to leave .	i have to see you .	i just wanted to see you .	just to see you .
see you later .	bye !	now wait a minute .	benjamin ?	benjamin ?	okay .
my name is david . what is my name ?	<UNK>.	sebastian . is that you ?	benjamin .	umyu name .	mrs . robinson .
my name is john . what is my name ?	what ?	mrs . robinson .	benjamin .	my name 's john .	mrs . robinson .
who is bill clinton ?	i do n't know .	no one knows . it 's his brother .	you know her .	well it 's not just me .	just some friends .
is sky blue or black ?	a car accident .	oh it 's a lovely name .	do you know what it is ?	blue .	blue .
does a cat have a tail ?	maybe .	yeah . i suppose so .	yes .	it does n't seem .	nope .
how many legs does a cat have ?	four .	ten .	i do n't know .	not too big .	i lived here .
what is the color of water ?	white .	oh it 's down .	i do n't really know where it is .	blue .	it 's like a song .
how much is two plus two ?	well do n't worry about it .	what is it ?	i do n't know .	eight .	oh i do n't know .
how much is ten minus two ?	yeah .	oh that .	not very .	twenty .	just about .

Table 7: Cornell S output responses for various name combinations.

## 6 Future Work

From the results presented in Section 5.2 the conclusion that the Transformer model simply doesn't perform as well as the seq2seq model could be drawn, however there are several reasons for the qualitatively worse results of the Transformer model trained with dialog data. First of all, the parameter space of the seq2seq model trained by [Vinyals and Le, 2015] is much bigger than the base variant of the Transformer that was used in this work. Secondly, as it has been discussed in Section 5.1, the Transformer trained on OpenSubtitles data seemed to stagnate which could be due to a too high learning rate or other hyperparameters not set correctly. Because of this it didn't achieve the point of convergence on the dataset and produced a lot of generic and safe responses. Since the same Transformer model and same hyperparameters were used as in the original work, which was tuned for NMT, it might very well be the case that the model would perform better with other hyperparameter configurations. Hyperparameter tuning is a general issue that subsequent research should focus on. Trying out larger Transformer variants for the OpenSubtitles dataset is also an important future direction.

Furthermore, it might be the case that the current standard loss function used has an even worse effect if used with the Transformer model. As discussed in Section 3.4.2, there are fundamental problems with this loss function and this is further proved in Section 5, where models that overfit the dataset and thus their validation loss increases, generate better and more diverse responses. Because of wrong hyperparameter settings or in consequence of the Transformer model not having a large enough parameter space, it couldn't even come close to overfitting the larger OpenSubtitles dataset, which is a good explanation of the safe and generic responses produced.

In addition to the ideas presented above to make the transformer model work better for conversational modeling, several ideas are proposed in the following sections towards solving some of the issues presented in Section 3.4. In Section 6.1 the loss function issue is tackled. Then, a conversational model that takes into account the passage of time is described in Section 6.2. Finally, further ideas related to constructing better conversational models are presented in Section 6.3.

### 6.1 Ideas Towards Solving The Loss Function Issue

Since human-like conversation is basically artificial general intelligence (AGI), it is extremely difficult to tackle it up front. A complex model would have to be constructed and a lot of knowledge included about the world in meaningful ways. This section will instead focus on augmenting encoder-decoder models with techniques that could remedy the loss function issue presented in Section 3.4.2. I propose that a multitude of features and priors should be taken into consideration when modeling conversations. Since maximizing the log probability is essentially the same as maximizing the probability of a reply given an utterance, there should be other prior probabilities that the reply can be conditioned on. This would address the issue of having various different responses for one source utterance, since differentiation between them by using other priors would become possible. A number of similar augmentations have been proposed before [Li et al., 2016a, Xing et al., 2017a, Zhou et al., 2017, Choudhary et al., 2017]. They all try to feed additional information into seq2seq models, like personality, mood and topic categories. While they all show that

generated responses from these models are somewhat more diverse than those outputted by a basic seq2seq model, I still believe that the conversational models are somewhat ambiguous and more priors should be used.

While at the birth of the seq2seq architecture conversational models were trained on utterance-reply pairs or by concatenating previous turns into one source sequence, there have been many efficient techniques proposed that can take into account conversation history, which are discussed in Section 3.1.1 and Section 3.3.1. Conversation history is indeed one of the most important priors that should be taken into account, since a response to an utterance is usually grounded on information presented in previous turns. However, this problem is mostly solved by the aforementioned techniques.

I propose several priors on which a conversational model should be conditioned. More specifically, the speaker and addressee of each utterance and the mood of the speaker at the time of saying the utterance. Taking into account the persona of the speaker and addressee is important, since what one says in general depends on one's personality and past experiences and also on who one talks to. This is already a good starting point, because it differentiates between different answers to same questions. However, even for a single person the reply to the question *How are you?* might be completely different depending on whether the person is happy, sad, annoyed, etc. By taking into account this mood prior, the data that the conversational model is trained on can be further disambiguated. Ideally, a dataset could be created where for each source sequence and set of priors only one correct output sequence exists. This would assure that the generated replies are diverse and not just an average of all possible responses. All of these priors and features can be represented by embeddings similar to word embeddings, which can be learned during training. However, as more features are added, more data is needed, so that model has sufficient examples from which it can learn what it *means* to be in a mood or talk like a specific person. This is a problem with the Cornell Movie-Dialog dataset, which is annotated with speaker-addressees, but it is too small and the model can't generalize from so few examples.

There are still problems with the above mentioned approach. Intuitively the speaker and addressee representations should have a lot of parameters, since they have to capture what makes one different from other people, which depends on all of one's past experiences. This is a lot of information to capture into a vector, for example to represent a single word 100-1000 dimensional vectors are used and since a lot of information (many words) are needed to describe a person, it would make the representation huge. There are two problems with these huge representations. The network would be very difficult to train because of limited computational resources and current datasets don't offer enough examples from which the network can learn such a huge space of parameters.

In order to combat these issues I propose a different approach. The priors previously mentioned are still used, however personality representations should be kept to learnable sizes. The intuition behind this is that people generally don't differ that much in basic world knowledge and conversational style. Thus, these representations could be kept smaller and instead the construction of a representation which captures general world-knowledge, language-knowledge and conversation-knowledge should be pursued. An example for world-knowledge would be the color of the sky, language-knowledge is about learning the meaning of words and conversation-knowledge is about

learning to answer with yes/no to yes-or-no questions for example. The advantage of this representation is that it could also be trained on non-conversational data with unsupervised methods.

There is one final prior that needs to be taken into account to truly capture the conditioning space of conversations. Take for example the reply *I got hit by a car*. to the question *How are you?*. Obviously, this response has little to do with the persona of the speaker or any general knowledge about the world. Rather the response is simply conditioned on outside factors, which can be temporary. Temporary, because after a week has passed from the accident in the previous scenario, the person will probably not reply with this answer, since *How are you?* is a question about the present. There are some ways in which these outside factors could be dealt with. For example the person’s speaker representation could be slightly changed for this specific response, or a new representation that tries to encode outside factors could be used. The most simple approach would be to just cut out conversations grounded on outside factors from the dataset. Unfortunately, all of these methods require a dataset that is labeled with labels that show whether the response is based on some external factor, which can probably only be done manually and to my knowledge such a dataset does not exist currently.

The actual specifics of how such representations could be integrated with encoder-decoder models can have many variations of course, however this work does not go into details.

## 6.2 Temporal Conditioning and Memory

An important part of conversations is timing and the involvement of both speakers. Current chatbot models are trained in such ways that they will only emit one response instantly after they receive the user’s utterance. To make chatbots more human-like I propose an additional term to the loss function, which is based on the temporal delay between an utterance and a reply. Essentially the chatbot has to generate a reply and also guess how much time should it wait before emitting the reply. Twitter-style datasets usually have such annotations so the implementation of this feature should be straight-forward.

The time-delay can be represented as a vector of probabilities over time frame categories. For example the model could have 10% confidence that the reply should be delayed by 0 to 10 seconds and 90% confidence that it should be delayed by 10 to 20 seconds. Such a representation is effective because a simple softmax can be used to get the probabilities of time frame categories and backpropagation can be extended by comparing the predicted time-delay vector with the one-hot ground-truth timing vector. Not only would this addition help achieve a more human-like delay (which could also be tunable) in the chatbot’s responses, but it would allow a conversational model to periodically generate new utterances by itself, based on the conversation history. Regardless of whether the user has inputted an utterance or not, the conversation history can be fed into the model after the chatbot emitted an utterance, and a new utterance can be generated if the model thinks that it is necessary to further the conversation. With the time delay mechanism, this process can go on indefinitely, since the generation of the new utterance will only be considered after the time delay for the current utterance has passed. This feature would make the chatbot even more human-like and engaging, by furthering the conversation without user interaction.

Another concept tied to temporal conditioning is real-time model updates. Basically, at each

turn the user’s utterances can be backpropagated through the network, based on the conversation history. This would make the model update its weights and prior representations as the conversation goes on, thus remembering the dialog. The technique could either replace or complement the need for taking into account long conversation histories with hierarchical models. It would be especially useful for real-world deployment of chatbots, where users expect a chatbot to be able to remember what they said 1000 utterances ago, which can’t be achieved solely through a hierarchical model. Through these real time updates of the parameters the model is capable to encode and learn new information about the user, which acts as memory. For example for a new user the persona representation is not known yet, however by interacting with the user the model can learn information specific to that user and it can encode it into the representation with real-time backpropagation.

### 6.3 Additional Ideas

In this final section several further ideas are presented that could be used to build better conversational models. Reinforcement learning has achieved impressive results recently for game-like tasks where there is a clear end-goal, without any human supervision [Silver et al., 2017]. Some approaches to adapt RL to the field of conversational modeling have been discussed in Section 3.3.3. My idea would be to train 2 encoder-decoder based conversational agents and then let them talk to each other. Furthermore, one of the chatbots would have the goal of getting a specific response from the other bot. If in some number of turns the response is uttered, then both bots could get a positive reward. In this environment several goals can be designed that would help further train the chatbots without any data or supervision. The downside of this idea is that hand-crafted goals have to be designed, which should capture important conversational properties.

New neural architectures should also be tried for the conversational domain, for example models that were described in Section 3.3.4. Furthermore, an interesting technique called neural architecture search [Zoph and Le, 2016] has been proposed to build new types of recurrent cells and even entire convolutional architectures. Since there already exist a plethora of different computational blocks used in conversational models, these could simply be input to the neural architecture search model, which could then generate an architecture that would presumably be better suited for the task of conversational modeling.

## 7 Conclusion

A brief overview of the history of chatbots has been given and the encoder-decoder model has been described in detail. Afterwards an in-depth survey of scientific literature related to conversational models, published in the last 3 years, was presented. Various techniques and architectures were discussed, that were proposed to augment the encoder-decoder model and to make conversational agents more natural and human-like. Criticism was also presented regarding some of the properties of current chatbot models and it has been shown how and why several of the techniques currently employed are inappropriate for the task of modeling conversations. Furthermore, preliminary experiments were run by training the Transformer model on two different dialog datasets. The performance of the trainings was analyzed with the help of automatic evaluation metrics and by comparing output responses for a set of source utterances. Finally, it was concluded that further, more detailed experiments are needed in order to determine whether the Transformer model is truly worse than the standard RNN based seq2seq model for the task of conversational modeling. In addition to presenting directions and experiments that should be conducted with the Transformer model, several ideas were presented related to solving some of the issues brought up in the criticism that was given earlier in the paper. These ideas are an important direction for future research in the domain of conversational agents, since they are not model related, but rather try to solve fundamental issues with current dialog agents. Continuation of this work will focus on trying to make open-domain conversational models as human-like as possible by implementing the ideas presented.

## References

- [Akasaki and Kaji, 2017] Akasaki, S. and Kaji, N. (2017). Chat detection in an intelligent assistant: Combining task-oriented and non-task-oriented spoken dialogue systems. *arXiv preprint arXiv:1705.00746*.
- [Apple, 2017] Apple (2017). Siri. <https://www.apple.com/ios/siri/>. Accessed: 2017-10-04.
- [Ba et al., 2016] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.
- [Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- [Barone and Sennrich, 2017] Barone, A. V. M. and Sennrich, R. (2017). A parallel corpus of python functions and documentation strings for automated code documentation and code generation. *arXiv preprint arXiv:1707.02275*.
- [Bengio et al., 2003] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.
- [Bordes et al., 2016] Bordes, A., Boureau, Y.-L., and Weston, J. (2016). Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.
- [Bottou, 2010] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT’2010*, pages 177–186. Springer.
- [Britz, 2015] Britz, D. (2015). Recurrent neural network tutorial. <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/>. Accessed: 2017-10-09.
- [Carpenter, 2017] Carpenter, R. (2017). Cleverbot. <http://www.cleverbot.com/>. Accessed: 2017-10-04.
- [Chen and Manning, 2014] Chen, D. and Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.
- [Cheng et al., 2016] Cheng, J., Dong, L., and Lapata, M. (2016). Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.
- [Chiu et al., 2017] Chiu, C.-C., Lawson, D., Luo, Y., Tucker, G., Swersky, K., Sutskever, I., and Jaity, N. (2017). An online sequence-to-sequence model for noisy speech recognition. *arXiv preprint arXiv:1706.06428*.

- [Cho et al., 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- [Choudhary et al., 2017] Choudhary, S., Srivastava, P., Ungar, L., and Sedoc, J. (2017). Domain aware neural dialog system. *arXiv preprint arXiv:1708.00897*.
- [Danescu-Niculescu-Mizil and Lee, 2011] Danescu-Niculescu-Mizil, C. and Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87. Association for Computational Linguistics.
- [Das et al., 2017] Das, A., Kottur, S., Moura, J. M., Lee, S., and Batra, D. (2017). Learning cooperative visual dialog agents with deep reinforcement learning. *arXiv preprint arXiv:1703.06585*.
- [Dodge et al., 2015] Dodge, J., Gane, A., Zhang, X., Bordes, A., Chopra, S., Miller, A., Szlam, A., and Weston, J. (2015). Evaluating prerequisite qualities for learning end-to-end dialog systems. *arXiv preprint arXiv:1511.06931*.
- [Eric and Manning, 2017] Eric, M. and Manning, C. D. (2017). A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. *arXiv preprint arXiv:1701.04024*.
- [Facebook, 2017] Facebook (2017). The babi project. <https://research.fb.com/downloads/babi/>. Accessed: 2017-10-13.
- [Feng et al., 2017] Feng, Y., Zhang, S., Zhang, A., Wang, D., and Abel, A. (2017). Memory-augmented neural machine translation. *arXiv preprint arXiv:1708.02005*.
- [Gehring et al., 2017] Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.
- [Ghazvininejad et al., 2017] Ghazvininejad, M., Brockett, C., Chang, M.-W., Dolan, B., Gao, J., Yih, W.-t., and Galley, M. (2017). A knowledge-grounded neural conversation model. *arXiv preprint arXiv:1702.01932*.
- [Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- [Google, 2017] Google (2017). Google assistant. <https://assistant.google.com/>. Accessed: 2017-10-04.
- [Goyal et al., 2017] Goyal, K., Neubig, G., Dyer, C., and Berg-Kirkpatrick, T. (2017). A continuous relaxation of beam search for end-to-end training of neural sequence models. *arXiv preprint arXiv:1708.00111*.

- [Gu et al., 2016] Gu, J., Lu, Z., Li, H., and Li, V. O. (2016). Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.
- [Harris, 1954] Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.
- [Havrylov and Titov, 2017] Havrylov, S. and Titov, I. (2017). Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *arXiv preprint arXiv:1705.11192*.
- [He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [Henderson, 2015] Henderson, M. (2015). Machine learning for dialog state tracking: A review. In *Machine Learning in Spoken Language Processing Workshop*.
- [Hochreiter, 1998] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [Jean et al., 2014] Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2014). On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*.
- [Jena et al., 2017] Jena, G., Vashisht, M., Basu, A., Ungar, L., and Sedoc, J. (2017). Enterprise to computer: Star trek chatbot. *arXiv preprint arXiv:1708.00818*.
- [Joshi et al., 2017] Joshi, C. K., Mi, F., and Faltungs, B. (2017). Personalization in goal-oriented dialog. *arXiv preprint arXiv:1706.07503*.
- [Kaiser and Bengio, 2016] Kaiser, Ł. and Bengio, S. (2016). Can active memory replace attention? In *Advances in Neural Information Processing Systems*, pages 3781–3789.
- [Kaiser et al., 2017a] Kaiser, L., Gomez, A. N., and Chollet, F. (2017a). Depthwise separable convolutions for neural machine translation. *arXiv preprint arXiv:1706.03059*.
- [Kaiser et al., 2017b] Kaiser, L., Gomez, A. N., Shazeer, N., Vaswani, A., Parmar, N., Jones, L., and Uszkoreit, J. (2017b). One model to learn them all. *arXiv preprint arXiv:1706.05137*.
- [Kalchbrenner et al., 2016] Kalchbrenner, N., Espeholt, L., Simonyan, K., Oord, A. v. d., Graves, A., and Kavukcuoglu, K. (2016). Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*.

- [Kandasamy et al., 2017] Kandasamy, K., Bachrach, Y., Tomioka, R., Tarlow, D., and Carter, D. (2017). Batch policy gradient methods for improving neural conversation models. *arXiv preprint arXiv:1702.03334*.
- [Konstas et al., 2017] Konstas, I., Iyer, S., Yatskar, M., Choi, Y., and Zettlemoyer, L. (2017). Neural amr: Sequence-to-sequence models for parsing and generation. *arXiv preprint arXiv:1704.08381*.
- [Kottur et al., 2017] Kottur, S., Moura, J. M., Lee, S., and Batra, D. (2017). Natural language does not emerge ‘naturally’ in multi-agent dialog. *arXiv preprint arXiv:1706.08502*.
- [Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- [Lample et al., 2016] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.
- [Li et al., 2015] Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2015). A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.
- [Li et al., 2016a] Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., and Dolan, B. (2016a). A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.
- [Li et al., 2016b] Li, J., Miller, A. H., Chopra, S., Ranzato, M., and Weston, J. (2016b). Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*.
- [Li et al., 2016c] Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., and Jurafsky, D. (2016c). Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.
- [Li et al., 2017] Li, J., Monroe, W., Shi, T., Ritter, A., and Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.
- [Li et al., 2016d] Li, X., Mou, L., Yan, R., and Zhang, M. (2016d). Stalematebreaker: A proactive content-introducing approach to automatic human-computer conversation. *arXiv preprint arXiv:1604.04358*.
- [Lin et al., 2017] Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.
- [Lison and Bibauw, 2017] Lison, P. and Bibauw, S. (2017). Not all dialogues are created equal: Instance weighting for neural conversational models. *arXiv preprint arXiv:1704.08966*.
- [Lison and Tiedemann, 2016] Lison, P. and Tiedemann, J. (2016). Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *LREC*.

- [Liu et al., 2016] Liu, C.-W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., and Pineau, J. (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.
- [Lowe et al., 2017] Lowe, R., Noseworthy, M., Serban, I. V., Angelard-Gontier, N., Bengio, Y., and Pineau, J. (2017). Towards an automatic turing test: Learning to evaluate dialogue responses. *arXiv preprint arXiv:1708.07149*.
- [Lowe et al., 2015] Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.
- [Luong et al., 2015] Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- [Luong et al., 2014] Luong, M.-T., Sutskever, I., Le, Q. V., Vinyals, O., and Zaremba, W. (2014). Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.
- [Manning et al., 1999] Manning, C. D., Schütze, H., et al. (1999). *Foundations of statistical natural language processing*, volume 999. MIT Press.
- [Marietto et al., 2013] Marietto, M. d. G. B., de Aguiar, R. V., Barbosa, G. d. O., Botelho, W. T., Pimentel, E., França, R. d. S., and da Silva, V. L. (2013). Artificial intelligence markup language: A brief tutorial. *arXiv preprint arXiv:1307.3091*.
- [Microsoft, 2017a] Microsoft (2017a). Cortana. <https://www.microsoft.com/en-us/windows/cortana>. Accessed: 2017-10-04.
- [Microsoft, 2017b] Microsoft (2017b). Microsoft bot framework. <https://dev.botframework.com/>. Accessed: 2017-10-04.
- [Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- [Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- [Miller et al., 2016] Miller, A., Fisch, A., Dodge, J., Karimi, A.-H., Bordes, A., and Weston, J. (2016). Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*.
- [Miller et al., 2017] Miller, A. H., Feng, W., Fisch, A., Lu, J., Batra, D., Bordes, A., Parikh, D., and Weston, J. (2017). Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*.

- [Mitchell, 1998] Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- [Mnih et al., 2013] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- [Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
- [Nallapati et al., 2016] Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.
- [Olah, 2015] Olah, C. (2015). Understanding lstm networks. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Accessed: 2017-10-08.
- [opensubtitles.org, 2017] opensubtitles.org (2017). Opensubtitles. <https://www.opensubtitles.org/>. Accessed: 2017-10-08.
- [Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.
- [Parikh et al., 2016] Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.
- [Park et al., 2008] Park, Y., Patwardhan, S., Visweswariah, K., and Gates, S. C. (2008). An empirical analysis of word error rate and keyword error rate. In *INTERSPEECH*, pages 2070–2073.
- [Paulus et al., 2017] Paulus, R., Xiong, C., and Socher, R. (2017). A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.
- [Ramachandran et al., 2016] Ramachandran, P., Liu, P. J., and Le, Q. V. (2016). Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683*.
- [Rumelhart et al., 1988] Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.
- [Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.
- [Schuster and Paliwal, 1997] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

- [Sennrich et al., 2015] Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- [Serban et al., 2017a] Serban, I. V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., and Courville, A. C. (2017a). Multiresolution recurrent neural networks: An application to dialogue response generation. In *AAAI*, pages 3288–3294.
- [Serban et al., 2015] Serban, I. V., Lowe, R., Charlin, L., and Pineau, J. (2015). A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*.
- [Serban et al., 2017b] Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Ke, N. R., et al. (2017b). A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.
- [Serban et al., 2016] Serban, I. V., Sordoni, A., Bengio, Y., Courville, A. C., and Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784.
- [Serban et al., 2017c] Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A. C., and Bengio, Y. (2017c). A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301.
- [Shang et al., 2015] Shang, L., Lu, Z., and Li, H. (2015). Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*.
- [Shannon, 2017] Shannon, M. (2017). Optimizing expected word error rate via sampling for speech recognition. *arXiv preprint arXiv:1706.02776*.
- [Shao et al., 2017] Shao, Y., Gouws, S., Britz, D., Goldie, A., Strope, B., and Kurzweil, R. (2017). Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2200–2209.
- [Shazeer et al., 2017] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.
- [Shen et al., 2017] Shen, X., Su, H., Li, Y., Li, W., Niu, S., Zhao, Y., Aizawa, A., and Long, G. (2017). A conditional variational framework for dialog generation. *arXiv preprint arXiv:1705.00316*.
- [Silver et al., 2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

- [Silver et al., 2017] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.
- [Song et al., 2016] Song, Y., Yan, R., Li, X., Zhao, D., and Zhang, M. (2016). Two are better than one: An ensemble of retrieval-and generation-based dialog systems. *arXiv preprint arXiv:1610.07149*.
- [Sordoni et al., 2015] Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., and Dolan, B. (2015). A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*.
- [Sriram et al., 2017] Sriram, A., Jun, H., Satheesh, S., and Coates, A. (2017). Cold fusion: Training seq2seq models together with language models. *arXiv preprint arXiv:1708.06426*.
- [Srivastava et al., 2014] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.
- [Sukhbaatar et al., 2015] Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.
- [Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- [Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.
- [Tao et al., 2017] Tao, C., Mou, L., Zhao, D., and Yan, R. (2017). Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. *arXiv preprint arXiv:1701.03079*.
- [Tensorflow, 2017] Tensorflow (2017). Sequence-to-sequence models. <https://www.tensorflow.org/tutorials/seq2seq>. Accessed: 2017-10-08.
- [Tiedemann, 2009] Tiedemann, J. (2009). News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.
- [Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.
- [Vinyals and Le, 2015] Vinyals, O. and Le, Q. (2015). A neural conversational model. *arXiv preprint arXiv:1506.05869*.

- [Wallace, 2009] Wallace, R. S. (2009). The anatomy of alice. *Parsing the Turing Test*, pages 181–210.
- [Weizenbaum, 1966] Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.
- [Wen et al., 2016] Wen, T.-H., Vandyke, D., Mrksic, N., Gasic, M., Rojas-Barahona, L. M., Su, P.-H., Ultes, S., and Young, S. (2016). A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.
- [Werbos, 1990] Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.
- [Weston et al., 2015] Weston, J., Bordes, A., Chopra, S., Rush, A. M., van Merriënboer, B., Joulin, A., and Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.
- [Williams et al., 2017] Williams, J. D., Asadi, K., and Zweig, G. (2017). Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *arXiv preprint arXiv:1702.03274*.
- [Williams, 1992] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.
- [Wiseman and Rush, 2016] Wiseman, S. and Rush, A. M. (2016). Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960*.
- [Worswick, 2017] Worswick, S. (2017). Mitsuku. <http://www.mitsuku.com/>. Accessed: 2017-10-04.
- [Wu et al., 2016] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- [Xing et al., 2017a] Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., and Ma, W.-Y. (2017a). Topic aware neural response generation. In *AAAI*, pages 3351–3357.
- [Xing et al., 2017b] Xing, C., Wu, W., Wu, Y., Zhou, M., Huang, Y., and Ma, W.-Y. (2017b). Hierarchical recurrent attention network for response generation. *arXiv preprint arXiv:1701.07149*.
- [Yao et al., 2016] Yao, K., Peng, B., Zweig, G., and Wong, K.-F. (2016). An attentional neural conversation model with improved specificity. *arXiv preprint arXiv:1606.01292*.
- [Yao et al., 2015] Yao, K., Zweig, G., and Peng, B. (2015). Attention with intention for a neural network conversation model. *arXiv preprint arXiv:1510.08565*.

- [Yin et al., 2017] Yin, Z., Chang, K.-h., and Zhang, R. (2017). Deepprobe: Information directed sequence understanding and chatbot design via recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2131–2139. ACM.
- [Yu et al., 2017] Yu, Z., Black, A. W., and Rudnicky, A. I. (2017). Learning conversational systems that interleave task and non-task content. *arXiv preprint arXiv:1703.00099*.
- [Zhao et al., 2017a] Zhao, T., Lu, A., Lee, K., and Eskenazi, M. (2017a). Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability. *arXiv preprint arXiv:1706.08476*.
- [Zhao et al., 2017b] Zhao, T., Zhao, R., and Eskenazi, M. (2017b). Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.
- [Zhou et al., 2017] Zhou, H., Huang, M., Zhang, T., Zhu, X., and Liu, B. (2017). Emotional chatting machine: Emotional conversation generation with internal and external memory. *arXiv preprint arXiv:1704.01074*.
- [Zhu et al., 2017] Zhu, Q., Li, Y., and Li, X. (2017). Character sequence-to-sequence model with global attention for universal morphological reinflection. *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 85–89.
- [Zoph and Le, 2016] Zoph, B. and Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.

# AI Model Utilization Measurements For Finding Class Encoding Patterns

Peter Bajcsy\* Antonio Cardone Chenyi Ling  
 Philippe Dessauw Michael Majurski Tim Blattner  
 Derek Juba Walid Keyrouz

National Institute of Standards and Technology, Gaithersburg, MD 20899  
 {peter.bajcsy, antonio.cardone, chenyi.ling, philippe.dessauw, michael.majurski,  
 timothy.blattner, derek.juba, walid.keyrouz}@nist.gov

## Abstract

This work addresses the problems of (a) designing utilization measurements of trained artificial intelligence (AI) models and (b) explaining how training data are encoded in AI models based on those measurements. The problems are motivated by the lack of explainability of AI models in security and safety critical applications, such as the use of AI models for classification of traffic signs in self-driving cars. We approach the problems by introducing theoretical underpinnings of AI model utilization measurement and understanding patterns in utilization-based class encodings of traffic signs at the level of computation graphs (AI models), subgraphs, and graph nodes. Conceptually, utilization is defined at each graph node (computation unit) of an AI model based on the number and distribution of unique outputs in the space of all possible outputs (tensor-states). In this work, utilization measurements are extracted from AI models, which include *poisoned* and *clean* AI models. In contrast to clean AI models, the poisoned AI models were trained with traffic sign images containing systematic, physically realizable, traffic sign modifications (i.e., *triggers*) to change a correct class label to another label in a presence of such a *trigger*. We analyze class encodings of such clean and poisoned AI models, and conclude with implications for trojan injection and detection.

## 1 Introduction

The *motivation* of this work lies in the lack of interpretability and explainability of artificial intelligence (AI) models in security and safety critical applications. For

---

\*point of contact

instance, regular traffic signs and any physically realizable trigger modifications represent intended and hidden encoded classes in a classification AI model (e.g., a yellow sticky on top of a *STOP* traffic sign as a trigger changing the label from the intended *STOP* to the target *65 m/h* traffic sign classes [1]). Our lack of understanding of how classes are encoded in AI models for classifying traffic signs poses a safety threat in self-driving cars because AI models can contain such injected triggers causing misclassification. In addition to the application-specific motivation, methods for explainable AI are motivated in general by regulatory agencies, end users, decision makers, and engineers as they provide utilities for bias detection, trust in predictions, suitability for deployment, debugging, and recourse [2], [3].

We introduce the *terminology* used in this paper early on due to a varying usage of published terms in a broad spectrum of theoretical contributions to AI. We will refer to an AI model as a computation graph that (a) is a directed graph representing a math function and (b) consists of subgraphs. A subgraph is a subset of graph vertices (or graph nodes) connected with edges in the parent graph. Graph nodes of a computation graph are computation units (or graph components) that perform linear or non-linear operations on input data, (e.g., convolution, tangent hyperbolic activation function, and maximum value operation). In our work, the names of the AI models (or architectures) are adopted from literature since we are not creating any custom computation graphs. The input and output data at each computation unit are multidimensional arrays denoted as tensors. When an image from a class  $c$  flows through a computation graph, each computation unit generates real-valued tensors called class activations (the term is derived from an activation function that decides whether a neuron should be activated or not). A tensor generated by input images has dimensions reflecting a number of images (batch size), channels, rows, and columns. For a batch size equal to one, a tensor can be interpreted as a hyperspectral image. In our work, a class activation mapping is thresholding, and binarized channel values in one tensor are denoted as a tensor-state with rows  $\times$  columns of tensor-state values.

The *objectives* of this work are (1) to define utilization-based class encodings and AI model fingerprints, (2) to measure class encodings in architectures beyond small models (e.g., LeNet model with 60K parameters) and toy datasets, such as MNIST (Modified NIST dataset with 70K images of size  $28 \times 28$  pixels), and (3) to identify encoding patterns (motifs) that discriminate AI models without and with hidden classes (denoted as clean and poisoned AI models). Our *ultimate objective* is to identify and decompose AI model computation graphs into subgraphs (subnetworks) that serve specific detection, segmentation, classification, or recognition purposes. Building a library of subgraphs with semantic interpretations, such as a subgraph for wheel detection, creates opportunities to custom-design AI architectures for specific training datasets and avoid exploring a huge search space of graphs as done in the work of Ying et al. [4]. This objective is aligned with the search for visual patterns in a collection of visualizations of AI layers and neurons under the OpenAI Microscope project [5]. By understanding class encoding patterns, one can additionally benefit from reduced

AI model storage and inference computational requirements via more efficient network architecture search [4] with advanced hardware [6]. Furthermore, one can improve expressiveness of AI model architectures via design [7] and efficiency measurements [8] or one can assist in diagnosing failure modes [9].

This work addresses the *problems* of (a) designing utilization measurements of trained AI models and (b) explaining how training data are encoded in AI models based on those measurements. We approach the problems and address the three objectives as follows:

1. Define a class encoding by introducing a utilization measurement at each computation unit in an AI model computation graph.
2. Form a class encoding as a vector of utilization measurements at all computation units of an AI model.
3. Search for class encoding patterns by training and analyzing clean and poisoned AI models together with their training datasets, as well as by visualizing their patterns in AI model computation graphs, subgraphs, and tensor-state spaces.

Conceptually, utilization of any computation unit is related to a ratio of the number of different outputs (tensor-state values) activated by all training data points over the maximum number of possible outputs by the computation unit. Such utilization-based class encodings are useful as statistical representations of complex trained AI models for (a) classifying a large number of AI models as clean or poisoned, and (b) reducing the search space for understanding class's unique and overlapping patterns. We use a set of tensor-states at each graph node and for each training image as a baseline representation of one trained AI model. With such a baseline representation, one can visually validate correctness of any conclusions derived from utilization-based class encodings for varying class characteristics, application-specific datasets, and AI model architectures.

Figure 1 shows a high-level workflow for identifying discriminating patterns of class encodings in clean and poisoned AI models. The left side in Figure 1 illustrates "Training Dataset" consisting of clean (Class A) and poisoned (Class B) training images with a small red polygon denoted as a trigger (or poison). The left side could also be replaced with two clean classes as we aim to identify patterns of class encodings unique to each class and common for any two classes. Training images representing each class are inferred. During the inference of images from the same class, a vector of utilizations over all graph computation units is recorded and denoted as a class encoding. Differences in class encodings can be visualized by color-coded AI computation graphs to contrast class encodings (e.g., clean and poisoned or clean Class A and Class B - see the right side of Figure 1).

The key *challenges* in addressing the problems lie in (1) integrating theoretical knowledge about neural networks to define utilization-based class encodings, (2) computing class encodings within an allocated time (e.g., 15 min per AI model), with limited computational resources and over hundreds of thousands

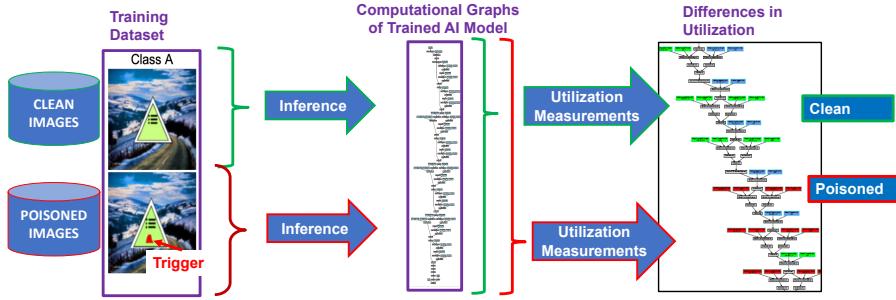


Figure 1: A high-level workflow for identifying utilization patterns in an AI computation graph of ResNet18 architecture for clean and poisoned classes

Table 1: Problems and their complexity challenges for AI models available from TrojAI Challenge, Rounds 1-4 [13]

Problems	Complexity Challenges
How to define AI model utilization?	tensor-states in AI models with $\approx 10^{12}$ parameters
How to characterize class encoding of each class via utilization of AI model computation units?	$\approx 10^5$ inferences per AI model
What AI model computation units are critical for class encodings?	$\approx 10^3$ AI model fingerprints

of training images, and (3) visualizing and interpreting class encoding patterns for a variety of AI model architectures at the granularity of AI model graphs, subgraphs, and tensor-states. All three challenges are intensified by increasing sizes of training datasets and by advancing complexities of AI architectures as enumerated with examples in Table 1. As of today, AI architectures are (a) very complex in terms of the number of parameters (from 60K parameters in LeNet model [10], to common networks having millions and billions of parameters, such as 160 billion reported by Trask et al. [11], and bleeding-edge networks with trillion-parameters in AI language models [12]), (b) very heterogeneous in terms of types of computation units in computational AI graphs, and (c) high dimensional in terms of data tensors generated by AI graph computation units.

The fundamental underlying *assumptions* of our approach lie in the fact that tensor-state statistics at each graph node from all activations with training images per class can gain us insights into a mathematical function defining the mapping between class-defining training images and class labels. The tensor-state statistics are quantified based on capacity versus utilization metrics. Such metrics capture the concept of a successful defense against backdoor attacks by graph pruning as reported by Liu et al. [14] and are assumed to reveal a presence

or absence of hidden classes (triggers or backdoor attacks). Although symbolic representations of subgraphs are still under investigation by Olah et al. [15], [16], we also assume that the utilization-based characterization of subgraphs may have a relationship with symbolic descriptions of image parts (e.g., subgraphs encode a traffic sign shape) and hence presence or absence of trojans can be detected by finding patterns in utilization-based color-coded graphs and subgraphs.

The main *novelties* of this work are in the definition, measurement design, and pattern searching in utilization-based clean and poisoned class encodings. The main *contributions* are in utilization measurement placements for a variety of AI architectures, and in explainable clean and poisoned AI models at the granularity levels of AI model graphs, subgraphs and tensor-states. Our work leveraged interactive Trojan and AI efficiency simulations enabled by the Neural Network Calculator tool [17] and web-accessible AI models generated for the TrojAI Challenge computer vision rounds [13].

The paper is organized as follows. Section 2 explains the relationship of this work to past efforts. Section 3 presents the theoretical underpinnings of AI model utilization measurements, the design reasoning, and the methodology for finding class encoding patterns in trained AI models. While Section 4 documents experimental data, numerical results, and visualizations, Section 5 provides an interpretation of the results in the context of trojan detection and injection. Finally, Section 6 summarizes lessons learned and outlines future work.

## 2 Related Work

The problem of explainable AI is very broad and the term *explainable* is still debated in philosophical texts [18] (“What is an Explanation?”). A comprehensive survey of explainable AI has been published by Arrieta et al. [2] and extensive teaching materials have been made available by Lakkaraju et al. [3]. Our approach can be related to “Explanation of Deep Network Representation” (roles of layers, individual units, and representation vectors) according to the Deep Learning-specific taxonomy presented by Arrieta et al. in [2], Fig. 11. Our utilization-based approach is inspired by exploring relationships between biological neural circuits and AI model computation graphs as discussed by Olah et al. in [15]. Next, the related work is presented with respect to the three formulated problems.

Our work on *defining utilization* is related to the past work on measuring neural network efficiency [8], [17], which is rooted in neuroscience and information theory. In the work of Schaub and Hotaling [8], neural efficiency and artificial intelligence quotient (aIQ) are used to balance neural network performance and neural network efficiency while inspired by the neuroscience studies relating efficiency of solving Tetris task and brain metabolism during the task execution [19]. In the work of Bajcsy et al. [17], an online simulation framework is used to simulate efficiencies of small-size neural networks with a variety of features derived from two-dimensional (2D) dot pattern data. In contrast to the previous work [8], [17], our theoretical framework defines and reasons about

class encodings, AI model fingerprints, and metrics for finding class encoding patterns for much more complex AI models and training datasets.

Following the categorization in the survey on interpreting inner structures of AI models [20], the *utilization measurements* can be related to concept vectors whose goal is to associate directions in latent space with meaningful concepts. In the work of Fong and Vedaldi [21] (Network 2 Vector) and Bau et al. [22] (Network Dissection), the distribution of activation maps at each convolutional unit as inputs pass through is used to determine a threshold. Threshold-based segmented activation maps are compared across concepts. In contrast to the previous work [21], [22], our utilization measurements are computed at all computation units in an AI model, the activation maps are binarized at zero, and statistics are computed over a distribution of tensor-states (including the binarized activation maps from convolutional units). Our approach does not use any inserted modules like in concept whitening [23] to align the latent space with concepts. Furthermore, our approach does not project class activation maps to create saliency maps [24], [25] in the input spatial domain, but, rather, it analyzes class activations in the tensor-state space.

Finally, following the categorization of approaches to understanding community (group or cluster) structure in AI models presented by Watanabe et al. [26], our overarching approach to *finding class encoding patterns* falls into the category “Analysis of trained layered neural networks” and combines two sub-categories: analysis of unit outputs and their mutual relationships and analysis of the influence on neural network inference by data. Overall, our approach can be related to modular partitioning [27], [28], and unsupervised disentanglement of a learned representation [29], [30]. In [27], [28], the authors search for local specializations of AI model subgraphs by using spectral clustering of AI model computation graphs and introducing two metrics, such as accuracy changes during neuron pruning ablation (neuron importance) and class-specific accuracy drop in a subcluster of neurons (input-feature coherence). In contrast to [27], [28], our clustering of computation units does not use “strong” and “weak” structural undirected connectivity of neurons as in spectral clustering, but, rather, repetitive co-occurrence patterns of utilization values in connected computation units. From the perspective of representation disentanglement, two studies by Locatello et al. [29], [30] perform extensive experiments on 14 000 models trained on eight datasets to conclude that well-disentangled models cannot be identified without supervision and the evaluation metrics do not always agree on what should be considered as “disentangled”. While we tacitly assume that high-dimensional data can be explained by lower dimensional semantically meaningful latent variables as by Locatello et al. [29], [30], we do not attempt to fully automate finding subgraphs (i.e., a human is always in the loop) to follow the conclusions by Locatello et al. [29], [30]. In addition, we do not aim at fully partitioning all AI model computation graphs into poly-semantic and mono-semantic subgraphs based on the poly-semantic and mono-semantic classification of neurons in subgraphs according to Olah et al. [15] and Räuker et al. [20].

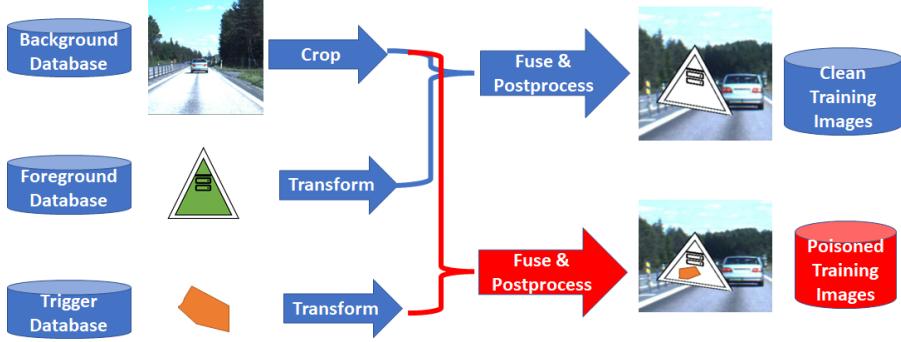


Figure 2: The process of creating training data with traffic signs. An example of a simulated triangular traffic sign and a polygon type of trigger.

### 3 Methods

In this section, utilization-based class encodings are defined by addressing the three key challenges listed in Section 1: (1) AI graph size and connectivity complexity, (2) component (graph node) heterogeneity, and (3) tensor dimensionality and real-value variability.

The utilization measurements of class encodings are defined by introducing tensor-states measured at the output of each component in AI computation graphs as training data points pass through the AI graph. The process of creating clean and poisoned training datasets is described next.

**Creation of clean and poisoned training datasets:** The training images for each class in TrojAI challenge (Rounds 1-4) are created according to Figure 2 by fusing and post-processing foreground and background images. Images of foreground traffic signs are constructed from images of real and simulated traffic signs. The background images are retrieved from existing road and city video sequences (e.g., citiscapes [31], KITTI 360 by Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago [32], and others [33]). A variety of images per traffic sign class is accomplished by changing parameters of crop, transformation, fusion, and post-processing operations as shown in Figure 2.

#### 3.1 Utilization-based Class Encoding: Definitions

**Clean and Poisoned AI models:** Let  $F_a : \mathbb{R}^m \rightarrow \{1, \dots, C\}$  refer to a trained AI model with architecture  $a$  that classifies two-dimensional  $m$ -variate images into one of  $C$  classes. When  $F_a$  is clean (denoted as  $F_a^\square$ ),  $F_a$  achieves a high classification accuracy over input images  $\vec{x}_i \in \mathbb{R}^m; i \in \{1, \dots, M\}$  where  $M$  is the number of pixels. When a clean  $F_a$  is poisoned by a trigger (denoted as  $F_a^\blacksquare$ ), there exists a function  $g : \mathbb{R}^m \rightarrow \mathbb{R}^m$  applied to input images from a source trigger class  $c_s$ , such that  $F_a(g(\vec{x}_i)) = c_t$ , where  $c_t$  is the target trigger class and  $c_t \neq c_s$ . Examples of clean images from source class, poisoned images from

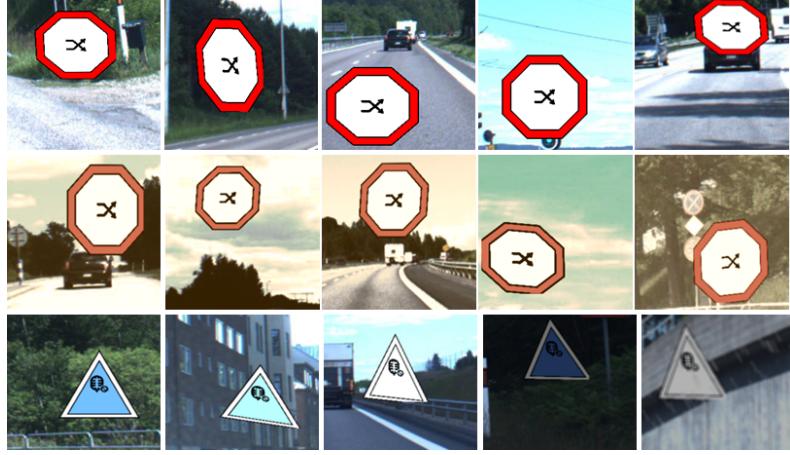


Figure 3: Instagram trigger: Examples of clean source class images  $\vec{x}_i$  (top), poisoned source class images  $g(\vec{x}_i)$  with  $g$  being a Kelvin Instagram filter (middle), and clean target class images  $F_a(g(\vec{x}_i)) = c_t$  (bottom).

source class, and clean images from target class are shown in Figure 3 (Instagram filter trigger) and in Figure 4 (Polygon trigger). For a pair of trained clean and poisoned AI models, labels for source class  $c_s$  and target class  $c_t$  are predicted with high accuracies according to the four equations below:

$$F_a^{\square}(\vec{x}) = c_s \text{ and } F_a^{\square}(g(\vec{x})) = c_s \quad (1)$$

$$F_a^{\blacksquare}(\vec{x}) = c_s \text{ and } F_a^{\blacksquare}(g(\vec{x})) = c_t \quad (2)$$

**AI computation graph:** A computation graph of a trained AI model  $F_a$  is denoted by  $G_a = \{V, E\}$  where  $V = \{v_1, v_2, \dots, v_{n(a)}\}$  are the  $n(a)$  computation units (or graph nodes or graph components) and  $E \subseteq V \times V$  are the edges. The unidirectional edges of a graph  $G_a$  are described by an adjacency matrix  $A \in \{0, 1\}^{n(a) \times n(a)}$  with  $A_{ij} = 1$  for all connected nodes  $v_i$  and  $v_j$ , and  $A_{ij} = 0$  for all other node pairs.

**tensor-state:** Each input image  $\vec{x}_i$  passes through  $G_a$  populated with trained coefficients. The input generates a tensor of output values at each computation unit (i.e., an activation map)  $v_j : \mathbb{R}^{D_j^{In}} \rightarrow \mathbb{R}^{D_j^{Out}}$ , where  $D_j^{In}$  and  $D_j^{Out}$  are the input and output dimensions of data at the computation unit  $j$ . The output values are binarized by zero value thresholding to form a tensor-state  $s_j(\vec{x}_i) = b(v_j(\vec{x}_i)) \in \{0, 1\}^{D_j^{Out}}$ ;  $b : \mathbb{R}^{D_j^{Out}} \rightarrow \{0, 1\}^{D_j^{Out}}$ . We refer to the graph location of  $v_j$  at which the output values are measured as a probe location. Figure 5 illustrates one tensor-state value for a specific ResNet101 computation graph, its specific graph node named layer1.2.conv2.weight, and one image from a predicted class  $c = 37$ . The example tensor-state  $(1, 64, 56, 56)$  is visualized as a set of 8 images with dimensions  $56 \times 56$  pixels, and the 64 bits (binarized outputs) are represented as 8 bytes.

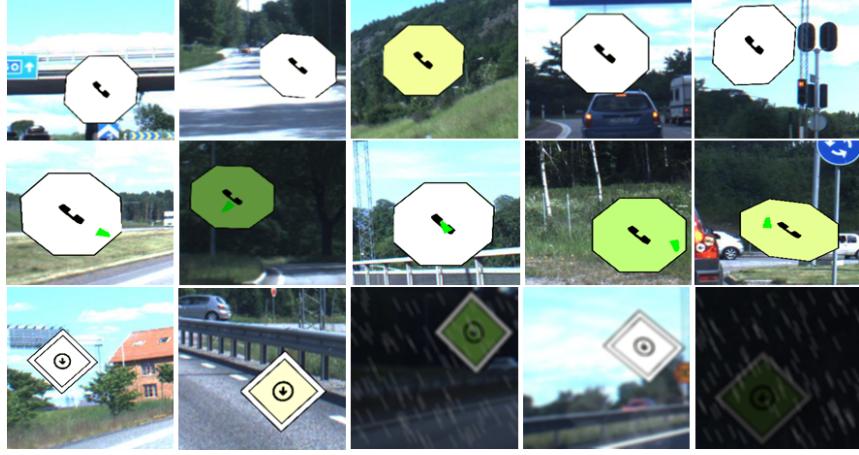


Figure 4: Polygon trigger: Examples of clean source class images  $\vec{x}_i$  (top), poisoned source class images  $g(\vec{x}_i)$  with  $g$  being a bright green polygon trigger (middle), and clean target class images  $F_a(g(\vec{x}_i)) = c_t$  (bottom).

**tensor-state Distribution:** Given a set of measured tensor-states  $\{s_j(\vec{x}_i)\}$  at a computation unit  $v_j$  for which  $F_a(\vec{x}_i) = c$ , let us denote  $Q_j(c) = \{q_{ij}(c)\}_{i=1}^{n_j}$  to be a discrete probability distribution function (PDF) over all tensor-state values, where  $n_j = 2^{D_j^{Out}}$  is the maximum number of available tensor-state values at the  $j$ -th computation unit  $v_j$ . The value of  $q_{ij}(c)$  is the sum of counts of unique tensor-state values  $count_{ij}$  invoked by all images  $i$  ( $\bigvee i \rightarrow s_j(\vec{x}_i)$ ) and normalized by the maximum number of available tensor values  $n_j$ . Figure 5 (bottom left) shows the histogram values  $count_{ij}$  computed from 5 366 576 unique tensor-state values over all 2 500 training images of *STOP pedestrian crossing* traffic signs. Based on the tensor-state dimensions (1, 64, 56, 56), one can establish the maximum number of predicted classes for such a node to be  $C_{layer1.2.conv2}^{MAX} = \frac{2^{64}}{56*56*2500} \approx 2.35 * 10^{12}$ ; a terascale count of traffic sign classes.

**Reference tensor-state Distribution:** For a class-balanced training dataset with similar class complexities, let us refer to  $P_j = \{p_{ij}\}_{i=1}^{n_j}$  as the uniform (reference) PDF over all states;  $p_{ij} = \frac{1}{2^{D_j^{Out}}}$ . The probabilities  $p_{ij}$  are associated with each state (index  $i$ ) and each computation unit (index  $j$ ) for each class  $c$ .

**Utilization:** We can compute a scalar utilization value  $\eta_j(c)$  for each class label  $c$  and a computation unit  $v_j$  from the count of measured states  $q_{ij}(c)$  and the state distribution  $Q_j(c)$  based on Equations 3-5. Equation 3 defines utilization  $\eta_j^{state}$  based on a deterministic view of states. In contrast, Equations 3 and 4 define utilizations  $\eta_j^H$  and  $\eta_j^{KLDiv}$  based on a probabilistic view of states by computing entropy  $H(Q_j)$  of a state distribution normalized by maximum entropy  $H_j^{max}$  or reference distribution  $P_j$ . The three utilization definitions yield value ranges  $\eta_j^{state} \in [0, 1]$ ,  $\eta_j^H \in [0, 1]$ , and  $\eta_j^{KLDiv} \in [0, \infty]$  per computation unit with an index  $j$ . For increasing utilization, the state- and entropy-based

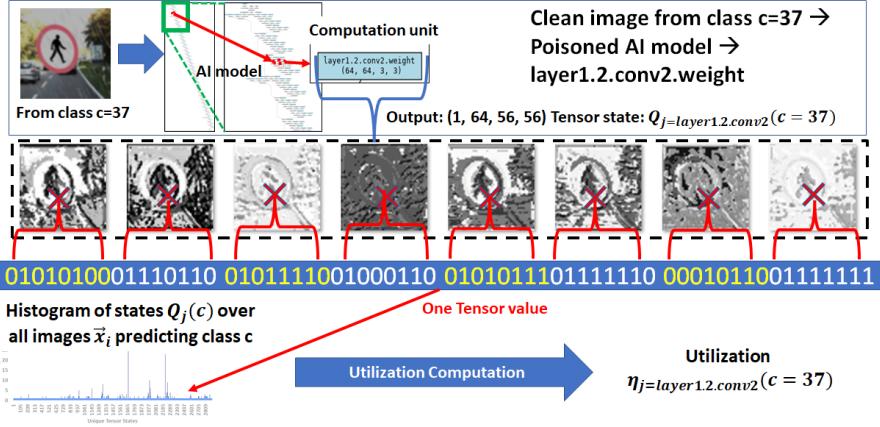


Figure 5: An example of a tensor-state and its one value derived from a ResNet101 computation graph (node layer1.2.conv.weight) contributing to a tensor-state distribution used for utilization computations.

measurements will increase while the Kullback–Leibler(KL) Divergence-based measurement will decrease since it measures non-utilization (or a deviation from the reference uniform distribution of tensor-states across all predicted classes). The KL Divergence-based measurement assumes that the maximum number of available states  $n_j$  is uniformly divided across all predicted classes (i.e., class encodings consume an equal number of available tensor-states).

$$\eta_j^{state} = \sum_{i=1}^{n_j} \frac{\text{count}_{ij}}{n_j} = \sum_{i=1}^{n_j} q_{ij} \leq 1 \quad (3)$$

$$\eta_j^H = \frac{H(Q_j)}{H_j^{max}} = \frac{-\sum_{i=1}^{n_j} (q_{ij} * \log_2 q_{ij})}{\log_2 n_j} \quad (4)$$

$$\eta_j^{KLDiv} = D_{KL}(Q_j \parallel P_j) = \sum_{i=1}^{n_j} (q_{ij} * \log_2 \frac{q_{ij}}{p_{ij}}) \quad (5)$$

The vector of utilization values for all AI computation units  $j \in \{1, \dots, n(a)\}$  is referred to as *a class encoding  $\vec{e}(c)$  for the class c*. The vector of utilization values from all classes  $c \in \{1, \dots, C\}$  if referred to as *a probe encoding  $\vec{r}(j)$  for the computation unit j*. A set of class encodings for  $c \in \{1, \dots, C\}$  ordered by the class label is denoted as *an AI model utilization fingerprint  $\mathbf{U}_a = \{\vec{e}(c = 1), \dots, \vec{e}(c = C)\}$* . An example of AI model fingerprint is shown in Figure 6 for ResNet101 architecture.

**Utilization Properties:** Utilization values are nondecreasing for increasing number of training data, number of predicted classes, decreasing AI model capacity. Experimental verifications of these utilization measurement properties can be found in Appendix B.

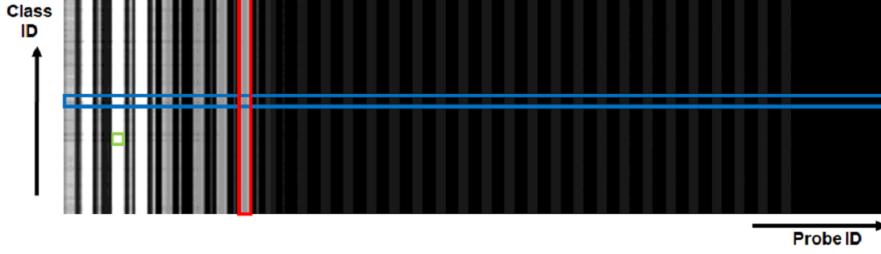


Figure 6: AI model utilization fingerprint  $\mathbf{U}_a$  for a trained poisoned architecture  $a$ , ResNet101, predicting 75 classes ( $C = 75$ ) and the source class  $c = 25$  being poisoned by an Instagram filter (shown in Figure 3) to misclassify traffic signs to a target class  $c = 34$ . The blue rectangle along a row shows a utilization-based class encoding  $\vec{e}(c)$ , the red rectangle along a column shows a utilization-based multi-class probe encoding  $\vec{r}(j)$ , and the small green square shows one utilization value  $\eta_j(c)$ .

### 3.2 Class Encoding: Measurements

**Utilization Measurement Workflow:** Following the theoretical definition, the utilization workflow steps are shown in Figure 7. The workflow starts with placing multiple measurement probes to collect the activation maps and follows the sequence of steps on the right side of Figure 7: record tensor-states, compute a histogram of tensor-states, derive class encoding for one class, and form an AI model utilization fingerprint. The placement of a measurement probe is after each computation unit.

**Computational Complexity of Utilization Measurements:** Following on the key challenges introduced in Section 1, the utilization measurements require significant computational resources for managing tensor-states in memory. The measurement involves building state histograms, computing the utilization values according to Equations 3 - 5 per computation unit of AI computation graph, and repeating the calculations over hundreds of computation units per graph while evaluating hundreds of thousands of images per AI model and thousands of trained AI models. The requirements to compute one utilization-based AI model fingerprint  $\mathbf{U}_a$  can be estimated based on Equations 6 (execution time  $T(\mathbf{U}_a)$ ) and 7 (Memory).

$$T(\mathbf{U}_a) = M \times \hat{T}(F_a(\vec{x}_i)) \quad (6)$$

$$\text{Memory} \leq \max_j(D_j^{Out}) \times M \times n(a) \quad (7)$$

where  $M$  is the number of input images for all  $C$  predicted classes,  $\hat{T}(F_a(\vec{x}_i))$  is the estimated average inference time per image,  $\max_j(D_j^{Out})$  is the maximum output cardinality of a computation unit  $v_j$  (max number of output nodes), and  $n(a)$  is the number of measurement probes in a graph for architecture  $a$ . For numerical examples, see Section 5.

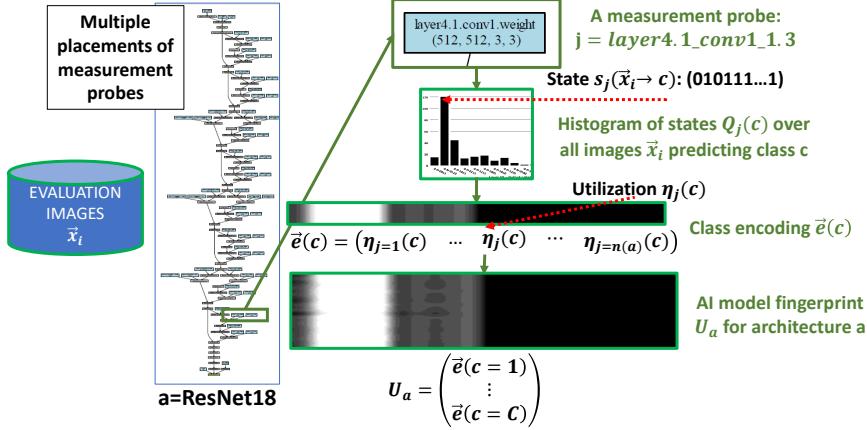


Figure 7: An example workflow for computing an AI model fingerprint for ResNet18 architecture from a set of evaluation images.

We approached the computational challenges by

- reducing the number of training images per class and building an extrapolation model,
- analyzing the AI model architecture designs to limit the number of probes, and
- modifying the KL Divergence computation according to [17] to reduce computations.

First, reducing the number of training images per class can be achieved via sampling and extrapolation modeling assuming that the classes are well represented in the tensor-state space with fewer samples (see Section 4).

Second, one can leverage a hierarchical structure of some AI computation graphs. Figure 8 shows a hierarchy of nodes, layers, and blocks that form an AI model. As AI model architecture designers define, combine, and connect computation units, AI computation graphs are partitioned into programming modules (i.e., methods, layers or blocks) that can be used for placing measurement probes. Since the programming modules represent a logical partition of a computation graph based on ad-hoc or exhaustive experimentations (i.e., the Neural Architecture Search problem [4]), they can define initial placements of measurement probes and lower memory and computational requirements for utilization computations.

Third, the KL Divergence computation in Equation 5 requires aligning measured and reference states, which is computationally expensive (for corner cases: if  $q_{ij} = 0$  then  $\eta_j^{KLDiv} = 0$  and if  $p_{ij} = 0$ , then  $\eta_j^{KLDiv} = 0$  because  $\lim_{x \rightarrow 0} (x * \log_2 x) = 0$ ). We assume that, on average, the uniform (reference)

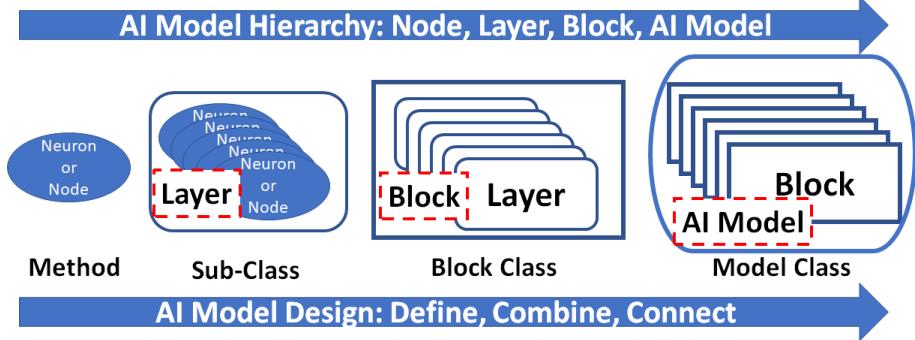


Figure 8: A hierarchical design of complex AI models as a logical partition of AI model computation graphs.

Table 2: Sets of utilization measurements

Data Set Index	AI Model	Evaluated with Class Images
Set 1	Clean	Clean
Set 2	Poisoned	Clean
Set 3	Poisoned	Source class + trigger

PDF is equally divided among predicted classes (efficient class encodings), which eliminates the need for aligning.

**Class-Specific Utilization Evaluation:** Utilization values can be measured by evaluating clean and poisoned trained AI models by clean and poisoned training images. We considered utilization measurements to be derived from the sets described in Table 2. Other evaluation options will be explored in the future.

### 3.3 Utilization-based Class Encoding: Finding Patterns

The problem of finding patterns is defined with respect to three granularity levels of AI models summarized in Table 3. The granularity levels are introduced to cope with the complexity of class encodings in AI models at the level of (a) individual computation units that encode tensor-states of images  $s_j(\vec{x}_i \rightarrow c)$ , (b) computation subgraphs that capture utilization motifs per class  $\vec{e}(c)$ , and (c) AI model fingerprints  $U_a$  that represent utilizations of all classes in a computation graph. The micro to macro granularity levels can be leveraged for hierarchical analyses of AI models as we are inspecting thousands of AI models in TrojAI challenge rounds, predicting 15 to 45 classes, using 5 to 20 computation graphs (i.e., architectures), and 2500 training images per class (see Table 1 listing complexity challenges).

Table 3 also lists the measurements and metrics at each granularity level. Measurements consist of unique tensor-states, utilization-based class encoding

Table 3: Granularity of finding patterns

Granularity	Measurements	Analyzed metrics
Computation unit (graph node)	Unique tensor-states per class	Distribution of common tensor-states
computation subgraph	Utilization class- encoding vector	Vector correlations of class encodings
computation graph (fingerprint of AI model)	Utilization matrix (class vs. probe)	Delta of utilization histogram bins

vectors, and utilization matrices. Metrics are applied to individual measurements or pairs of measurements to identify patterns, for instance,

- spatial overlaps of semantically meaningful image regions with tensor-state values (e.g., common blue sky versus class-unique traffic sign symbols in invoked tensor-states),
- partial similarities of multiple class encodings in the same AI model (e.g., encoding of traffic sign classes utilizing similar and dissimilar AI model computation subgraphs), and
- similarity of AI model utilization fingerprints in AI model collections (e.g., common utilization of multi-class encodings in multiple AI model architectures).

**Patterns detected in computation units:** In addition to the computational challenges associated with computing utilization as described in Section 3.2, one must address the visualization challenges for viewing multidimensional tensors. The challenge was approached by forming 8 images for one tensor-state of size  $(1, 64, 56, 56)$  in Figure 5 where the tensor-state was invoked by one input image at one of the ResNet101 computation units. Equation 8 provides the formula for calculating a total number of images representing tensor-states that could be visually inspected.

$$N_{\text{Images}}^{\text{Tensors}} = \sum_j^{n(a)} \left( \frac{D_j^{\text{Out}}}{8} \right) \times M \quad (8)$$

where  $M$  is the number of input images for all  $C$  predicted classes,  $D_j^{\text{Out}}$  is the number of outputs from a computation unit  $v_j$ , and  $n(a)$  is the number of measurement probes in a graph for architecture  $a$ . For example, for the ResNet101 computation graph with  $n(a) = 286$  utilization measurement probes and  $M = 100\,000$  images (40 predicted traffic sign classes represented by 2500 training images per class) and on average  $D_j^{\text{Out}} = 64$  dimensional outputs, one would need to inspect about  $286 \times \frac{64}{8} \times 100\,000 = 2.288 \times 10^8$  grayscale images.

To lower the number of images, we focus primarily on pairs of classes (clean and clean or clean and its corresponding poisoned classes). To simplify our visual

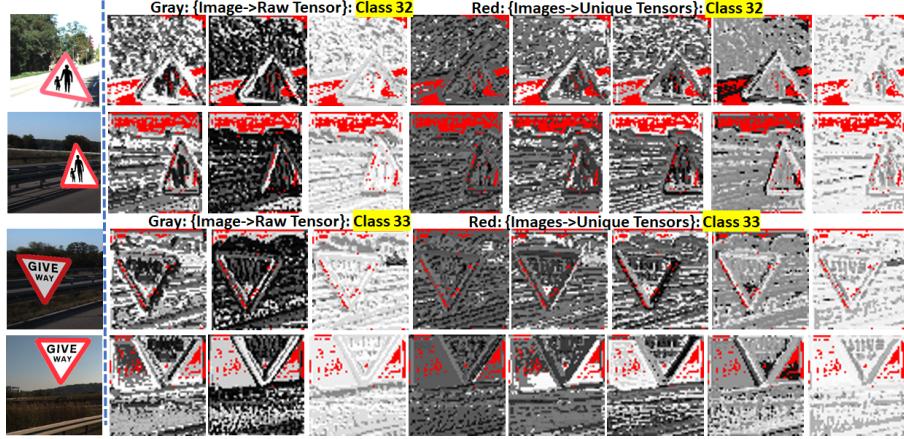


Figure 9: Visualization of tensor-state values in red for two images from two classes (Class 32 - *Parent and child* and Class 33 - *GIVE WAY* road signs) in layer1.2.conv2 of the ResNet101 model that occur more than 100 times in 2500 training images of the same class.

inspection, we look for all unique tensor-state values in all training images per traffic sign class with a frequency higher than a threshold. These high-frequency tensor-state values are then highlighted in each set of 8 images representing one tensor-state invoked by one training image as shown in Figure 9. Based on images like in Figure 9, one can derive conclusions about the presence of features in each image that are common across a training collection defining a class. For instance, Figure 9 (top rows) would indicate that the class *Parent and child* road signs contains many blue sky and saturated regions, as well as some key discriminating parts within the triangular road sign.

Figure 10 shows two images from class 32 with red dots overlaid at tensor-state values overlapping with class 33 (top two rows). The bottom two rows illustrate two images from class 33 with red dot overlaid at tensor-state values characterizing all images in class 32. By comparing Figure 9 and Figure 10, one can observe that both classes (1) have common tensor-state values corresponding to the blue sky and saturated regions and (2) do not overlap in tensor-state values defining the foreground traffic signs (*Parent and child* and *GIVE WAY*) except from a small number of white pixels. Note that a few tensor-state values in the red rim of each traffic sign are common to each class according to Figure 9, but they are not common to both classes as they differ in the shades of red. Following the classification of graph computation units as poly-semantic and mono-semantic neurons [15], [20], we can also conclude that *layer1.2.conv2* is poly-semantic since it is constructing common and unique class characteristics in the two traffic sign classes and passing them to the downstream graph computation units.

Another approach to inspecting the class encodings is via histograms of tensor-state values. Table 4 summarizes statistics of the number of unique tensor-state

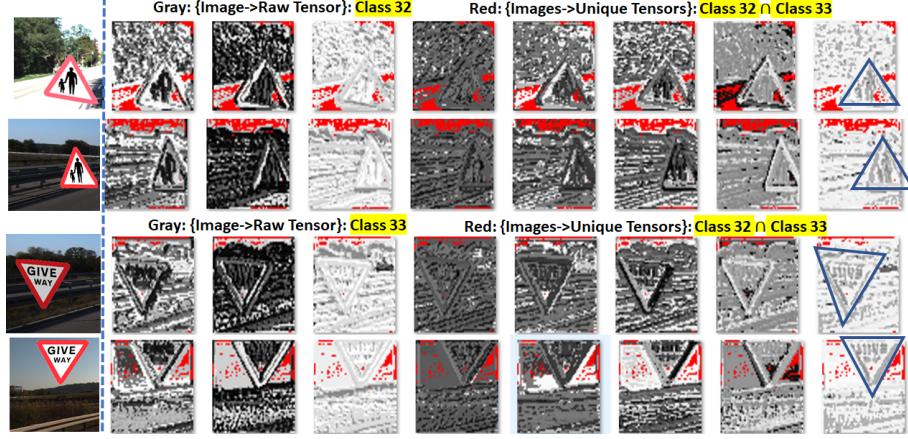


Figure 10: Visualization of tensor-state values in red for two images from two classes (Class 32 - *Parent and child* and Class 33 - *GIVE WAY* road signs) in layer1.2.conv2 of ResNet101 that occur more than 100 times in 2500 training images of one class and are present in two example images of the other class.

Table 4: Number of unique tensor-state values invoked by 2500 clean images and 2500 poisoned images with frequencies higher than 0, 1, 10, and 100 in layer1.2.conv2 of ResNet101 (image examples are shown in Figure 9). )

Num. unique state values	Class 32	Class 33
> 0	5 610 111	5 815 399
> 1	508 419	529 384
> 10	28 606	25 870
> 100	1476	3633
$\eta_{j=layer1.2.conv2}^{entropy}$	33.2	33.6

values in layer1.2.conv2 of ResNet101 AI model for two classes of training images shown in Figure 9. To scale down the visualization requirements on a histogram with more than 5.8 million bins, we can threshold the bins based on tensor-state value frequencies. Figure 11 shows the histogram visualization for the threshold value equal to 100 using Microsoft Excel. The frequency (count) along a vertical axis is shown on a logarithmic scale to accommodate the wide range of count values. The tabular and histogram visualizations allow to observe the number of tensor-state values, their frequencies, and overlapping characteristics of classes as illustrated in Figure 11. The histogram in Figure 11(right) illustrates how overlapping unique tensor-states between class 32 and class 33 with frequencies larger than 100 (horizontal axis) would have difference frequencies (vertical axis) in each of the classes defined by 2500 images per class. These overlaps can be explained by using the same background pool of images with characteristics on its own (e.g., blue sky, trees, roads).

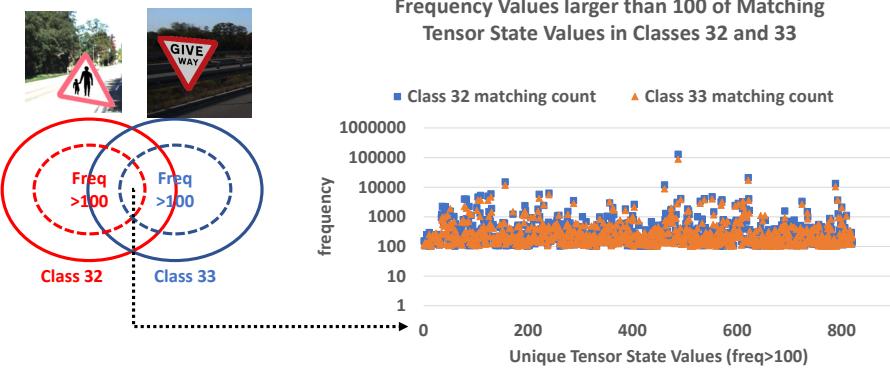


Figure 11: Left - Venn diagram of images and their unique tensor-states from two classes. The dashed circles refer to subsets of unique tensor-states that are invoked at least 100 times by all training images from each class. Right - Histograms of unique tensor-state values in layer1.2.conv2 of ResNet101 that occur more than 100 times in 2500 clean images shown in Figure 9. The dotted line connects the unique tensor-states in the histogram with the Venn diagram intersection region.

Note that in order to understand unique and overlapping class characteristics one would need to compute a much more complex Venn diagram for an AI model predicting  $C$  classes than the one in Figure 11 (left). While common and distinct characteristics of two classes can be compared in three different ways as illustrated in Figure 12, the number of comparisons for  $C$  classes would be  $2^C - 1$ , which quickly exceeds the limits of human inspections (e.g., for  $C = 40$  predicted traffic signs one would need to perform approximately  $10^{12}$  comparisons).

**Patterns detected in computation subgraphs:** The next level of granularity in explainable AI is to detect patterns in computation graphs according to the utilization-based class encodings as shown in Figure 1 (right). The motivation comes from the initial reports about semantically meaningful outputs of a group of computation units in AI models (e.g., partial curve detectors [15]) and a hypothesis that such groups would repeat to encode more complex curves. The problem lies in finding subgraphs in an AI computation graph (AI model) that are utilized the same way. This problem is known to be NP-hard (non-deterministic polynomial-time hardness) [34].

The baseline approach is to use a human visual inspection to identify class encoding patterns. Figure 13 illustrates the use of Torchvision [35], [36] for placing the measurement probes and DiGraph [37] for visualizing AI model computation graph with nodes color-coded according to the utilization values. Due to the visualization tradeoffs between rendered global and local information of very large graphs, complex connectivity, and heterogeneity of graph nodes, this visual inspection approach to finding patterns is not suitable for global comparisons of AI models (graphs).

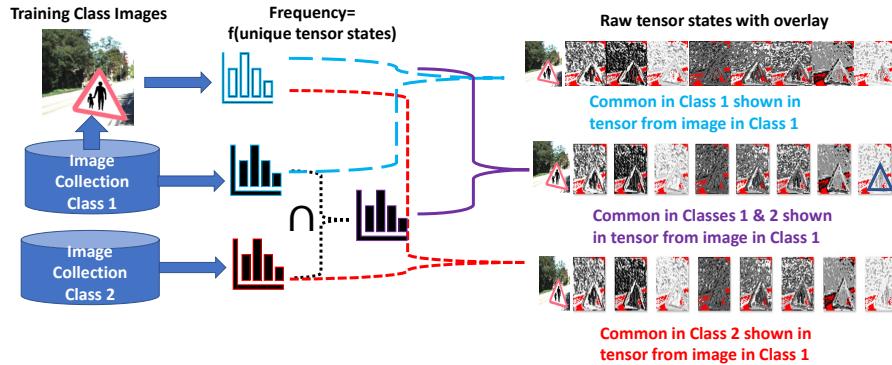


Figure 12: Conceptual comparisons of presence or absence of tensor-state values across two traffic sign classes. The right column shows in red the intersections of unique tensor-states from a single image in class 1 with the tensor-states coming from (a) a collection of images in the same class 1 (top right), (b) the intersection of tensor-states measured from two image classes (middle right), and (c) a collection of images in the other class 2 (bottom right).

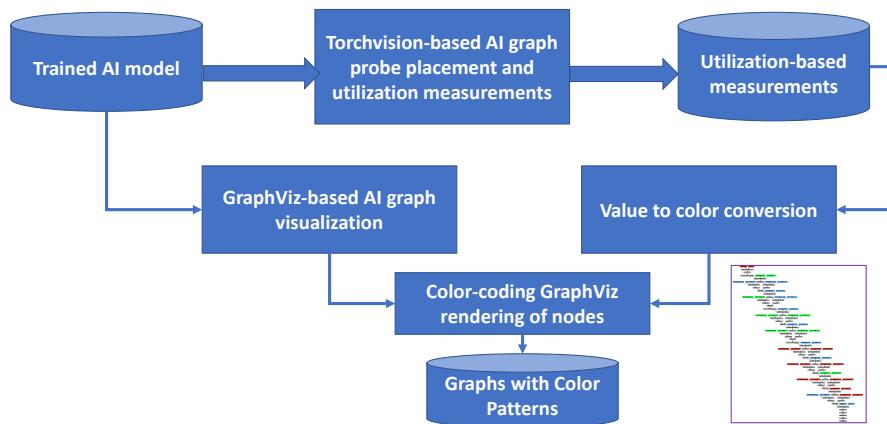


Figure 13: A workflow for pseudo-coloring graph nodes according to utilization measurements.

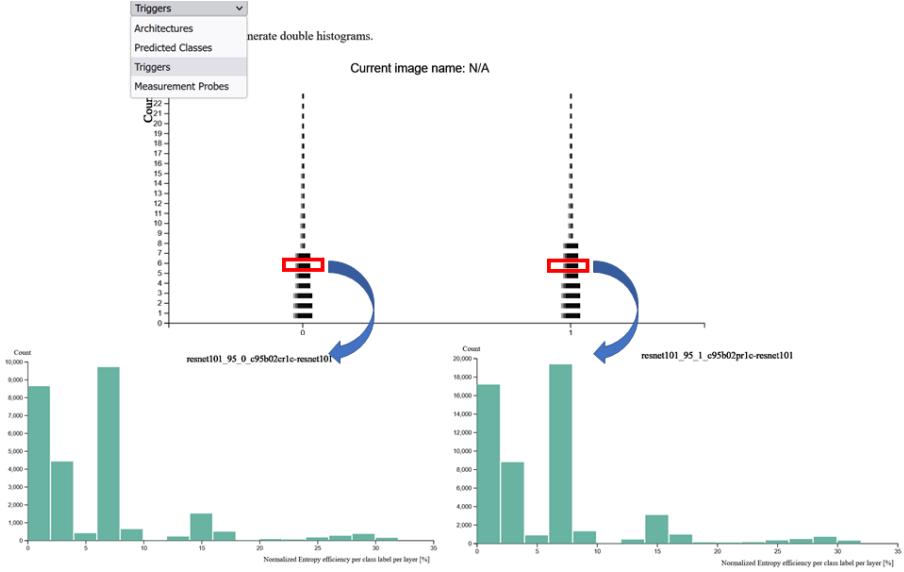


Figure 14: A pair-wise web-based comparison of AI model fingerprints. Top - histogram of AI model fingerprints with the drop-down menu for sorting based on attributes (top left). Bottom - two histograms from selected AI model fingerprints shown in red boxes.

**Patterns detected in computation graphs:** A fingerprint visualization shown in Figure 6 is a coarse-level summary of class encodings per AI model architecture. While analyzing a single AI model, one can immediately see similarity and dissimilarity of class encodings among classes along the vertical axis and utilization patterns of computation units in an AI computation graph along the horizontal axis. When analyzing multiple AI models, one can compare histograms of utilization values derived from multiple fingerprints. To support such comparisons, we have developed a web-based fingerprint comparison for collections of AI models that can scale to comparisons of thousands of AI models with different architectures (e.g., 1008 AI models with 16 architectures in Round 4 of TrojAI challenge). Model fingerprints can be sorted based on number of triggers, number of predicted classes, number of probes, and architecture types, and then compared via utilization histograms as illustrated in Figure 14.

## 4 Experimental Results

### 4.1 Experimental Datasets

We reused the software for generating the TrojAI Challenge datasets in Round 4 [38]. Due to a large variability of training images used for training AI models in TrojAI Rounds 1 to 4 (the training datasets are different for each of the

thousands of trained AI models per Round), we generated only two training datasets with a fixed generation seed while varying other parameters. In addition, we increased the number of predicted classes to  $C = 75$  and  $C = 95$  traffic signs. Following Figure 2, we created four training datasets (two training datasets  $\times \{C = 75, C = 95\}$ ) and trained with them three architectures including VGG13, SqueezeNet v1.1, and ResNet101 (once without triggers (clean), once with Instagram filter triggers, and once with polygon triggers). One trigger was inserted into a poisoned training dataset with a constant trigger fraction between clean and poisoned images equal to 0.5. Each AI model was trained three times with a different random training seed to explore the variability of class encodings. Each class was represented by 2500 color images, which amounted to  $75 \times 2500 = 187\,500$  and  $95 \times 2500 = 237\,500$  training images. The entire dataset consisted of 108 trained AI models (training datasets  $\times$  number of replicates  $\times$  number of architectures  $\times$  number of triggers =  $4 \times 3 \times 3 \times 3 = 108$ ).

## 4.2 Finding Patterns by Comparing Clean and Poisoned Classes

We describe the process of identifying graph locations for placing utilization measurement probes in Appendix C. As mentioned before, utilization measurements satisfy multiple properties that are experimentally supported in Appendix B. Similarly, we documented variability of utilization measurements in Appendix D and computational requirements in Appendix E.

To address the encoding complexity of clean and poisoned classes, we proceeded with the analyses from macro to micro granularity levels as documented in Table 3. We started with pattern detections in computation graphs first, next in subgraphs, and then in graph nodes. Our experiments are motivated by (a) evaluating our hierarchical utilization-based approach to classifying a large number of AI models and (b) understanding and validating the use of utilization measurements for this classification task at the tensor-state (micro) levels.

### 4.2.1 Patterns detected in computation graphs:

We illustrate the utilization patterns in class encodings for four trained models in Round 4 holdout dataset of TrojAI challenge with the parameters summarized in Table 5. The type of parametrization, the number of parameters, and their wide ranges of values represent a large space of possible trigger configurations. Such parametrizations are important for reducing a large search space of triggers by doing an experimental design for training poisoned AI models.

In our study, all four AI models are evaluated with clean images (i.e., Set 1 in Table 2). The four AI models are trained with different traffic signs, assigned randomly to 17 classes, and placed on top of randomly chosen backgrounds from cityscapes, kitti road, and kitti city image collections, and, therefore, the fingerprints cannot be compared by element-to-element.

Figure 15 shows a stacked histogram of an entropy-based utilization values for the four ResNet101 models. All four models have approximately the same

Table 5: Four trained models with the following parameters: architecture  $a = \text{ResNet101}$ , number of predicted classes  $C = 17$ , number of trojans  $g_i(\vec{x})$  per AI model  $\{0, 1, 2, 2\}$ , and trigger functions defined below.

Model ID	Model Type	Trigger 0	Trigger 1
142	Clean	$g_0(\vec{x}) = \vec{x}$	$g_1(\vec{x}) = \vec{x}$
235	Poisoned	$g_0(\vec{x}) = \text{Kelvin filter}$	$g_1(\vec{x}) = \vec{x}$
150	Poisoned	$g_0(\vec{x}) = \text{Gotham filter}$	$g_1(\vec{x}) = \text{Lomo filter}$
250	Poisoned	$g_0(\vec{x}) = 9\text{-sided polygon}$ of color [200, 0, 0]	$g_1(\vec{x}) = 4\text{-sided polygon}$ of color [0, 200, 200]

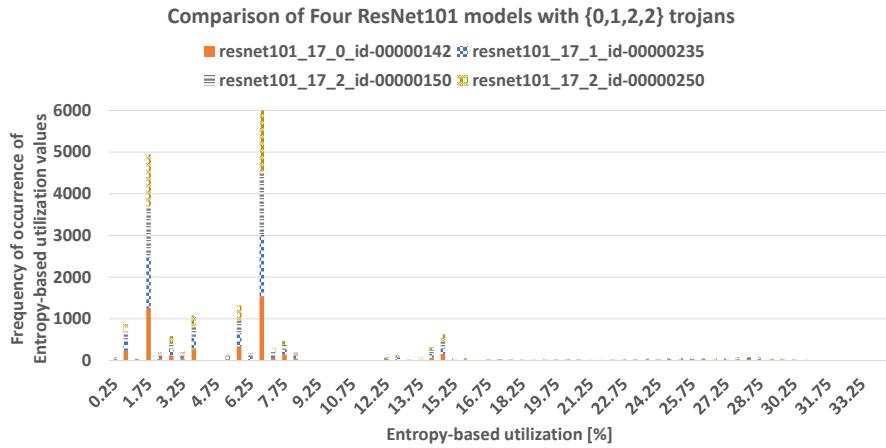


Figure 15: Stacked histogram of four ResNet101 models with zero, one, and two triggers (one model with two Instagram triggers and one with model two polygon triggers).

distribution of utilization values over all encoded traffic classes. However, as can be seen in Figure 16, there are utilization values in ranges  $[16.0, 18.0] \cup [18.5, 19.0]$  and  $[29.5, 31.5]$  that are present in the poisoned models but are missing in the clean model. The utilization values in  $[16.0, 18.0] \cup [18.5, 19.0]$  are measured at the computation units labeled as maxpool, conv1, bn1, and ReLU (maximum pooling, convolution, batch normalization, and rectified linear unit). The utilization values in  $[29.5, 31.5]$  come from layer1.2.conv2 and layer1.2.bn2 in all poisoned models. In addition, the values in  $[29.5, 31.5]$  are also measured in AI models poisoned with polygon triggers at the computation units labeled as layer1.1.conv2 and layer1.1.bn2. Based on this granularity-level analysis, one can focus on the identified subset of computation units to explain the clean versus poisoned class encodings.

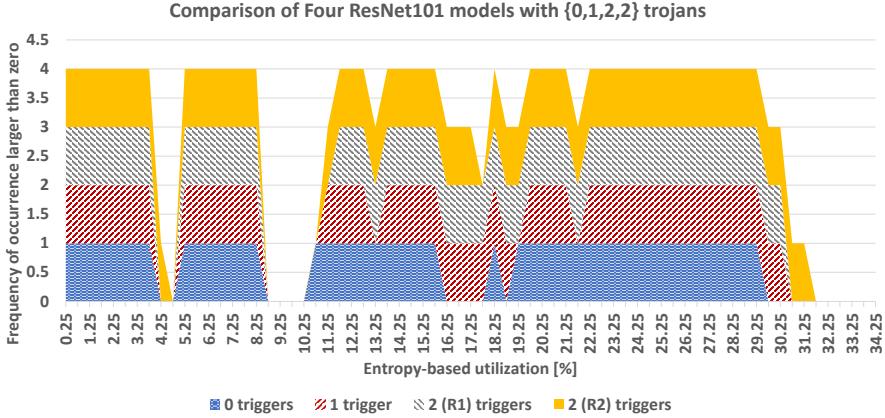


Figure 16: Comparison of four ResNet101 models with zero, one, and two triggers (two replicates denoted as R1 and R2).

#### 4.2.2 Patterns detected in computation subgraphs:

Class encodings in complex and large graphs can be rendered as vector or raster visualizations. Raster visualization faces a tradeoff between pixel resolution (graph node details) and display area (overall graph structure). After color-coding graph nodes based on utilization values, a raster image of ResNet101 architecture is around  $2000 \times 20000$  pixels and a vector representation in Adobe PDF truncates the content due to the large dimension. Figure 17 (left) shows an overview of the ResNet101 graph, which can be explored by zooming in and out as shown in the three zoomed out subgraphs (middle and right). In this class encoding of the 2500 traffic signs like the example shown in Figure 17 (bottom left), one can identify three distinct subgraphs that repeat in the utilization-based color-coded ResNet101 graph. We will focus primarily on Pattern 1 since it contains computational units (nodes) that are more utilized than those in Patterns 2 and 3. All presented results in this subsection are derived from AI models predicting 75 traffic sign classes.

In order to find utilization patterns, one is looking for repeating isomorphic subgraphs with the same utilization values assigned to all subgraph nodes. Such subgraphs encode class-specific parts in traffic sign classes, for example, several rotated curvatures in round shaped traffic signs using curve detectors [15] in mono-semantic subgraphs or multiple spatial- and intensity-based traffic sign characteristics in poly-semantic subgraphs. We hypothesize that a presence of physically realizable trojans will perturb utilization patterns.

Automated finding of isomorphic subgraphs is a NP-complete problem. Thus, we used visualization as the main approach to identifying utilization patterns. The visualization approach is supported (a) by extracting a graph structure (sometimes) available in Torchvision representation while following the workflow for pseudo-coloring graph nodes shown in Figure 13 and (b) by identifying high-

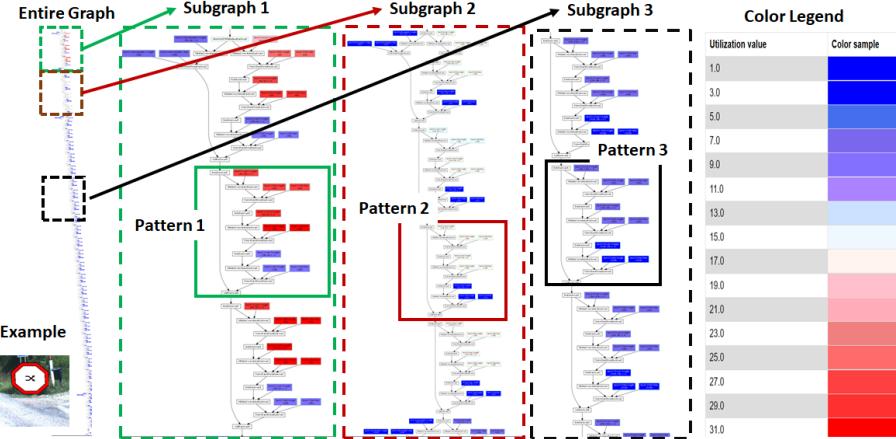


Figure 17: Three distinct patterns of a class encoding in the clean trained ResNet101 model. An example traffic sign image of the encoded class is in the lower left. The three subgraphs are zoomed out sections of the entire graph to illustrate three repeating patterns of colors with the color legend on the right.

frequency graph nodes of the same computation unit type via histograms. First, if the Torchvision representation of a AI model computation graph uses block classes, then one can use all graph nodes (computational units) with a block class as a candidate subgraph for finding a pattern. Second, one can leverage automatically computed histograms of graph nodes for ranking the graph nodes based on their frequency as candidates in repeating subgraph patterns. Figure 18 shows histograms of trace-based tree computation units based on block class names (left) and sub-class names (right) of ResNet101 architecture.

Encodings of clean classes in computation subgraphs: Figure 19 shows invariant utilization patterns for (a) two clean classes  $c = \{61, 63\}$  in the same trained AI model and (b) one class  $c = 63$  in two randomly initialized and trained AI models (Rep. 1 and Rep.2). The traffic signs are similar in triangular shape and red-white-black traffic colors, and dissimilar in the triangular orientation and black symbols inside triangles. The invariance of utilization patterns indicates that the uniqueness of each class encoding lies in the set of tensor-states while the utilization of computation units is constant (or the number of unique tensor-states per computation unit remains approximately constant in order to encode a class).

On the other hand, when traffic signs that are characterized by fewer unique properties than in Figure 19, for example the signs shown in Figure 20, subsets of training images with common properties within a class will be encoded at various computation units to improve classification accuracy and robustness to a few unique characteristics. Such distributed class encodings will yield to varying number of unique tensor-states at each computation unit and hence varying utilization. Figure 20 illustrates how varying color will cause perturba-

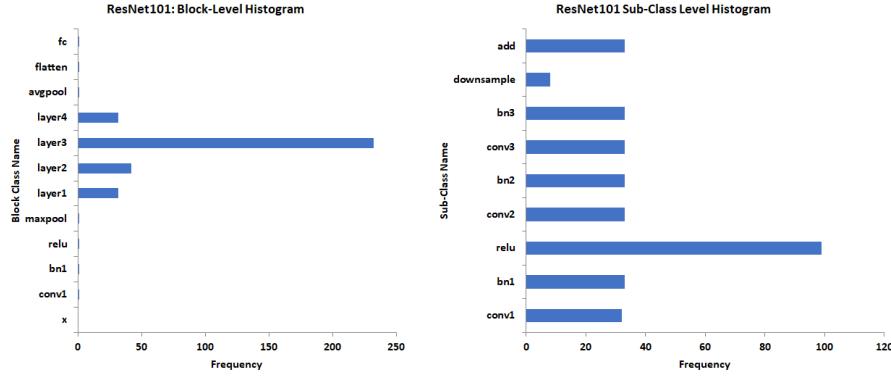


Figure 18: Histograms of block class names and sub-class names of ResNet101 architecture modules

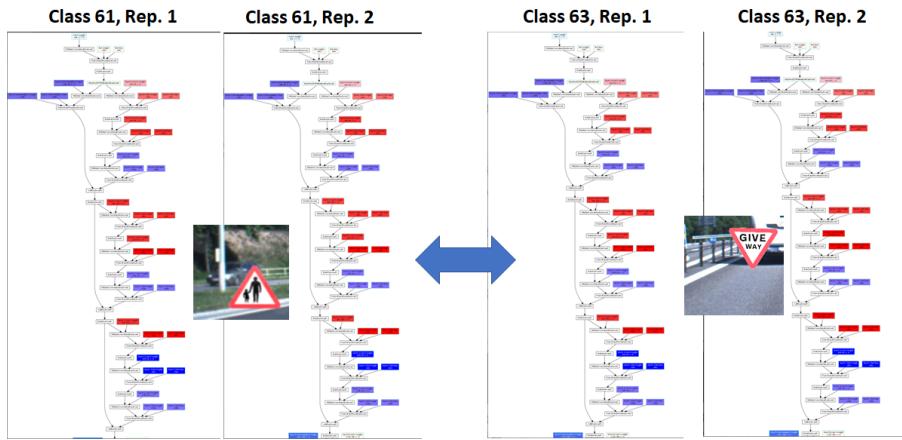


Figure 19: Comparison of class encodings for two triangular traffic signs with constant color and triangular shape in two trained replicate AI models (ResNet101 architecture). The color legend is the same as in Figure 17.

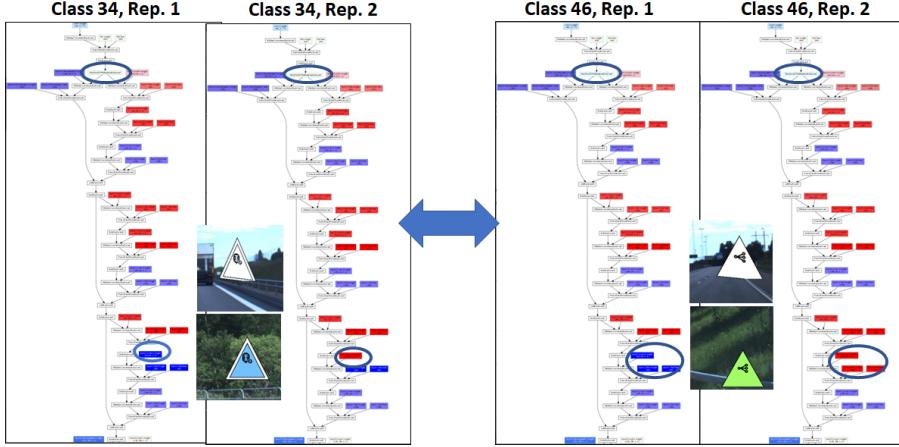


Figure 20: Comparison of class encodings for two triangular traffic signs with varying color and constant triangular shape in two trained replicate AI models (ResNet101 architecture). The color legend is the same as in Figure 17.

tions of entropy-based utilization patterns in maxpool (circled at the top) and layer1.2.conv2, layer1.2.bn2.weight and layer1.2.bn2.bias (circled at the bottom) for two replicate retrained AI models (randomly initialized) and two similar traffic signs. The two sample training images highlight only the color variability and the presence or absence of the white rim (left vs right).

Encodings of clean versus poisoned classes in computation subgraphs: Figure 21 shows the comparison of clean class encoding  $c = 25$  (left) and two replicate class encodings of  $c = 25$  with Kelvin Instagram filter as a trigger (middle and right) in the ResNet101 architecture. Based on the AI model fingerprint analyses in Section 4.2.1, Instagram filters and polygons as triggers present themselves in the initial maxpool, conv1, bn1, and ReLU computation units. Varying utilization (different from the clean class encoding) can be observed in Figure 21 with the circles enclosing maxpool2d, ReLU, conv1.weight, layer1.0.conv1.weight, layer1.0.conv2.weight, layer1.0.bn2.weight, and layer1.0.bn2.bias. The color coding goes from dark blue to dark red or from 1% to 31% of entropy-based utilization (see the color legend in Figure 17).

Regarding the subgraph pattern 1 shown in Figure 17, the trigger of Kelvin Instagram filter type breaks the pattern between layer1.1 and layer1.2 as highlighted with two dash-line rectangles in Figure 21. Since the Kelvin Instagram filter reduces the color spectrum to the earth tones of green, brown, and orange, this will reduce the number of unique tensor-states and, hence, reduce the utilization of some computation units.

It should be noted that patterns in subgraphs could also be studied for groups of traffic sign classes. For instance, one can group traffic signs by shapes to identify subgraphs that serve the same purpose but are parametrized differently. This type of pattern analyses must overcome the memory challenges as a larger

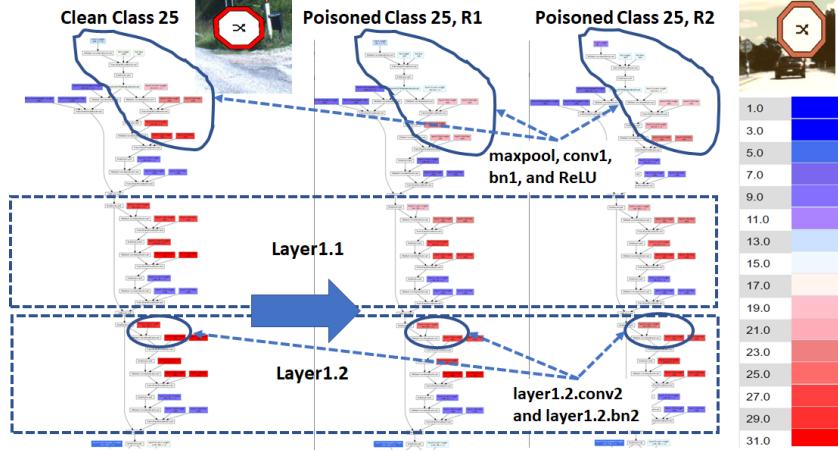


Figure 21: Comparison of a clean class encoding evaluated with clean images (left) and of poisoned class encodings in two trained replicate AI models (ResNet101 architecture) evaluated with poisoned images (middle and right). The circles show the variability of utilization in the initial graph nodes and layer1.0 in two poisoned class encodings. The rectangles show the utilization pattern change between clean and poisoned class encodings.

number of classes implies a larger number of training images and an even larger number of unique tensor-states to keep track of. Based on our experiments, we could evaluate up to four classes together or 10 000 training images of size  $256 \times 256$  pixels, and their corresponding unique tensor-states.

#### 4.2.3 Patterns detected in computation units:

Similar to Figures 9 and 10, we compared the tensor-states characterizing clean and poisoned classes in Figures 22 and 23. The comparison of clean and poisoned classes is shown for the same *STOP pedestrian crossing road* sign with or without applied Kelvin Instagram filter as a trigger. Figure 22 (top two rows) illustrates that the common tensor-state values within a clean class correspond to sky, parts of a road without shadows, and several pixel clusters inside the traffic sign. After applying the Kelvin Instagram filter, Figure 22 (bottom two rows), the common tensor-state values within a poisoned class are dominated by the earth tones of green, brown, and orange, which leads to a merger of semantically distinct regions, such as sky, parts of the road, and interior of the traffic sign. The image in the lowest row of Figure 22 has a significant number of red pixels suggesting that it consists of many features common across all poisoned images.

The objective of Figure 23 is to visualize with red pixels any common tensor-states across clean and poisoned classes. All four rows in Figure 23 show almost no red pixels except from a few pixels from the yellowish tree and from a red rim of the traffic sign in the top row left image. Since the Kelvin Instagram filter

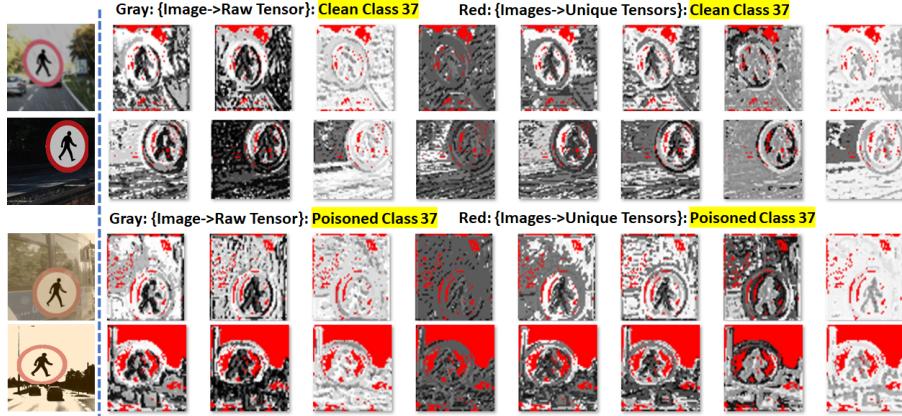


Figure 22: Visualization of tensor-state values in red for two sample clean (top two rows) and poisoned (bottom two rows) images from the same class in layer1.2.conv2 of ResNet101 that occur more than 100 times in 2500 clean images (top two rows) or in 2500 poisoned images (bottom two rows).

affects every pixel in a training image, the overlap of high-frequency tensor-state values between clean and poisoned images is only 35 tensor-state values and almost none in the area of the *STOP pedestrian crossing* traffic sign. In other words, although perceptually the areas of clean and poisoned traffic signs are very similar, the features characterizing each class as generated by the computation unit layer1.2.conv2.weight are completely different. Furthermore, since the Kevin Instagram filter blurs pixel values but makes their color more similar to each other, there are less unique tensor-state values in poisoned images than in clean images due to blur but more tensor-state value with high values due to color similarities.

Another approach to inspecting the class encodings is via histograms of tensor-state values. Table 6 summarizes statistics of the number of unique tensor-state values in layer1.2.conv2 for clean and poisoned training images in poisoned ResNet101 AI models. To scale down the visualization requirements on a histogram with 5.4 million bins, we threshold the bins based on the tensor-state value counts. Figure 24 shows the histogram visualization for the threshold value equal to 100 using Microsoft Excel. The frequency (count) along a vertical axis is shown on a logarithmic scale to accommodate the wide range of values. The horizontal values are sorted by frequencies.

The tabular and histogram visualizations allow us to observe the differences in the number of tensor-state values between clean and poisoned images as a function of unique tensor-state frequencies. The utilization values reflect the differences in the numbers of unique states and the distributions of their frequencies. While clean images give rise to more unique tensor-states than poisoned images ( $5\,366\,576 - 3\,734\,889 = 1\,631\,687$ ), they are also characterized

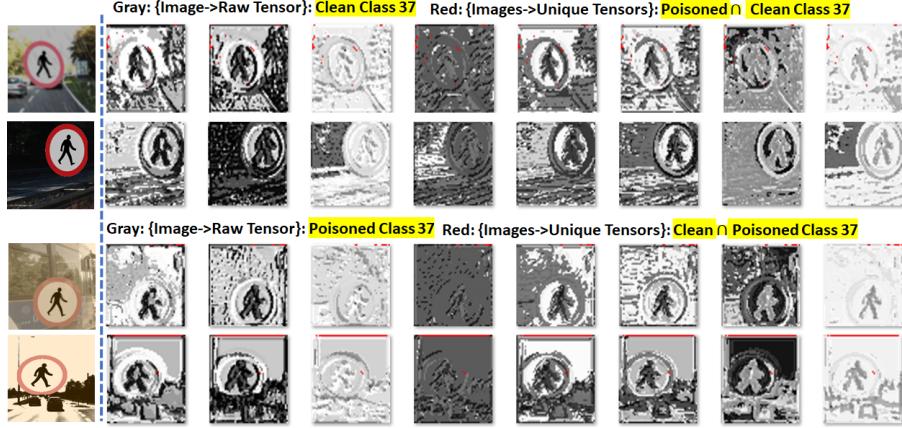


Figure 23: Visualization of tensor-state values in red for two sample clean (top two rows) and poisoned (bottom two rows) images from the same class (*STOP pedestrian crossing road signs*) in layer1.2.conv2 of ResNet101 that occur more than 100 times in both 2500 clean and 2500 poisoned images.

Table 6: Number of unique tensor-state values invoked by 2500 clean images and 2500 poisoned images with frequencies higher than 0, 1, 10, and 100 in layer1.2.conv2 of ResNet101 (image examples are shown in Figure 22).

Num. unique state values	Clean	Poisoned
$> 0$	5 366 576	3 734 889
$> 1$	531 183	645 297
$> 10$	35 047	57 985
$> 100$	1750	3633
$\eta_{j=layer1.2.conv2}^{entropy}$	33	30.6

by lower frequencies of tensor-states as shown in Table 6. We observe in Figure 24 that both distributions of sorted clean and poisoned tensor-states follow the same trend and hence the number of unique tensor-states becomes the key factor for a utilization value. This explains why the encodings of clean images yield a higher utilization  $\eta_{j=layer1.2.conv2}$  in the ResNet101 AI model than the encodings of poisoned images (see Table 4, last row).

## 5 Discussion

Trojan injections and detections: The examples shown in Figures 23 and 24 have some implications on trojan injections and detections. Two classes of images (clean and poisoned) with identical semantic labels according to a human visual inspection are encoded with completely different tensor-state representations. This evidence is presented for one of many Instagram filters as a trigger and at

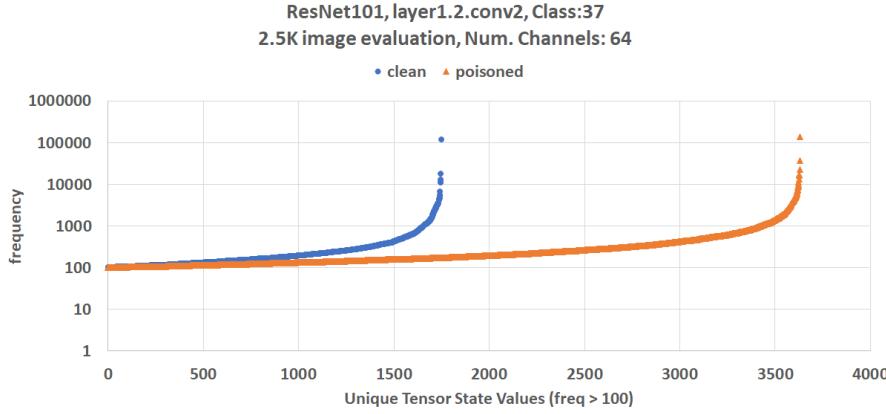


Figure 24: Histograms of unique tensor-state values in layer1.2.con2 of ResNet101 that occur more than 100 times in 2500 clean images (blue circles) and in 2500 poisoned images (orange triangles). Unique tensor-states are sorted by the frequency but are not matched across clean and poisoned images. Example images are shown in Figure 22.

one of many computation units in one of many AI model architectures. While we need additional studies to generate more class encoding examples and their dependencies on trigger types, we hypothesize that such relationships between training data and AI model can improve robustness of trojan injections and detections.

Assuming that trojans can be defined in a very large space of spatial and color image configurations, one can search for pairs of visually insignificant (or imperceptible) image changes and non-overlapping class encodings in order to improve robustness and convergence of trojan injections. Finding non-overlapping class encodings is related to the choice of AI model architecture (i.e., the choice of computation units and their connectivity) and can be framed as a type of a network architecture search (NAS) problem. If image classification training data and AI model architectures are understood in terms of utilization-based class encodings, then trojans can be designed in the image space to be encoded in underutilized computation units during training. On the other hand, trojan detectors can become more efficient in predicting poisoned AI models by first identifying computation units of interest, such as with pruning approaches [14], [39], [40], [41]. Trojan detectors can then be specialized by investigating subgraphs that (a) encode classes of trojans and (b) make the trojan search space for any detector much smaller.

Understanding overlapping characteristics of class encodings: We have studied overlapping characteristics of two clean classes or clean and poisoned classes. There are additional challenges in exploring overlapping characteristics of multiple encoded classes due to the much more complex Venn diagram for an AI

model predicting a number of classes (equal to  $C$ ). The challenges are not only in a larger number of visualizations, but they are also in the large cost of matching states across all combinations of classes and their lists of unique tensor-states at the order of several millions (see Table 6). Furthermore, given the cost of generating “explainable” class encodings and its purpose of delivering trusted predictions, one may be limited by the size of training data and complexity of AI models in the future. While explanations of class predictions have a ceiling on information content [42], the AI models do not have a ceiling on complexity. In other words, explanations must compress AI model complexity at a higher rate with the increasing model complexity [42]. Therefore, one would select only a few AI models with datasets to be fully characterized and trusted.

Natural trojans: We hypothesize that the tensor-states for clean and poisoned classes can explain why trojan detectors can be misled by natural trojans. Natural trojans in AI models are informally defined as those that yield a high false positive rate in predicting poisoned AI models. If one traffic sign class is trained on images characterized by two distinct groups, for instance, red and blue color traffic signs, then (a) trojan detectors might incorrectly rely on a correlation between the presence of trojans and a multi-modal distribution of training data for a single class, and (b) trojans can be injected by relabeling output labels for one group and retraining the model.

Detectability of trojans in poisoned AI models: While we showed a minimum tensor-state overlap between clean and poisoned class encodings in Figure 23 for semantically equivalent traffic signs of *STOP pedestrian crossing* sign, it is not clear whether detecting encodings of hidden classes is computationally feasible without a priori knowledge about types of triggers, types of foreground, and types of background by the trojan detector designers. One can view trojan detection as an outlier detection problem [43] and aim at establishing probably approximately correct (PAC) style learning bounds on the outlier detection under general assumptions. However, such limited guarantees on a successful trojan detection pose vulnerability risks as reported by Sun et al. [44] by demonstrating a patch attack in backdoored “broken” classifiers. Furthermore, Goldwasser et al. [45] has shown two frameworks of planting undetectable backdoors with incomparable guarantees (i.e., finding a trojan cannot be solved in polynomial time) for a class of ReLU networks. This topic of trojan detectability remains an open research problem.

Evaluation data for better explainability: Table 2 shows three types of evaluation data. Our evaluation sets did not contain pairs of clean and poisoned images that would differ only by the trigger. In other words, the background image and the fusion parameters were always different. One could understand the clean versus poisoned class encodings even better if the sets had matching pairs of clean and poisoned images. It would also be possible to explore the patterns in poisoned AI models with images from clean classes and with injected triggers.

Evaluation of AI model explainability: While we have shown three methods for explainable AI at graph, subgraph, and tensor-state granularity levels, we have not done quality evaluations of AI model explainability via model parameter

and data randomization tests [25]. It remains to be demonstrated how illustrated tensor-states, utilization patterns in subgraphs, and utilization distributions in multiclass prediction models actually reflect a true explanation of the traffic sign prediction.

Estimation of maximum number of classes that can be encoded: Given the number of training image per class and the tensor dimensions of a computation unit  $v_j$ , it is possible to compute the maximum number of classes that can be uniquely encoded by a graph unit  $v_j$ . This was calculated for the example tensor-state  $(1, 64, 56, 56)$  to be  $C_{layer1.2.conv2}^{MAX} = \frac{2^{64}}{56*56*2500} \approx 2.35 * 10^{12}$  (a terascale count of traffic sign classes). This number is purely theoretical since 2500 images of one type of a traffic sign would not fully represent the entire spectrum of imaged traffic signs in practice. In addition, it is not clear how the graph connectivity would be incorporated into the estimation.

Computational constraints on utilization measurements: Equations 6 and 7 summarize the formulas for evaluating computational challenges. For example, the computational cost of inferencing  $M = 2500$  images with  $n(a) = 286$  probes in  $a = \text{ResNet101}$  takes on average 24.46 minutes ( $\widehat{T}(F_a(\vec{x}_i) = 0.587)$ s while the memory consumption can reach up to 140.6 GB for one model from the TrojAI Challenge [46] (AI models in Round 4).

In this case, the input image size  $\vec{x}_i$  was used to approximate output tensor size  $\max_j(D_j^{Out})$  to be  $256 \times 256 \times 3 = 196\,608$  Bytes (Sample images are cropped to  $224 \times 224 \times 3$  in Round 4). The average time estimate was obtained experimentally on CPU: AMD Ryzen Threadripper 3970X 32-Core Processor; MemTotal: 264 GB and GPU: NVIDIA GeForce; MemTotal: 246 GiB. Measuring utilizations requires a processing time proportional to 1008 AI trained models, hundreds of computation units in each of the 16 included AI architectures, 2500 images per class, and between 15 and 45 predicted traffic sign image classes. These numerical values indicate our current hardware is limited to handling about 10,000 tensor-states in memory for  $a = \text{ResNet101}$  (Virtual Memory= 562.3 GB according to Equation 7).

## 6 Summary

We have introduced the concept of AI model utilization for the purpose of (a) delivering explainable AI models at graph, subgraph, and tensor-state granularity levels, and (c) accelerating injection and detection of poisoned AI models. We defined a mathematical framework for computing three deterministic and statistical AI model utilization metrics. Appendix A describes the relationship of these metrics to other theoretical approaches that have described AI models. Furthermore, we implemented a suite of tools for measuring utilizations of each computation unit in a computation graph and visualized the utilization measurements as matrices (AI model fingerprints), color-coded graphs, and a sequence of images representing a multidimensional array.

Specifically, we explained the utilization-based class encodings for clean and poisoned classes from the TrojAI Challenge (Rounds 1-4) [13]. We concluded that

while clean and poisoned images can clearly be classified into the same semantic traffic sign category, a poisoned AI model would have completely independent tensor-states for clean versus poisoned traffic sign images (see Figure 22 versus Figure 23). In addition, the tensor-state values observed as common to all images defining a class come from foreground and background image regions. Note that the importance of both types of regions for accurate prediction has been reported by Xiao et al. citeXiao2020. The utilization-based subgraphs for clean and poisoned classes illustrated that the utilization patterns changed (see Figure 21) and could become a coarse indicator of tampering with training data and distributing a poisoned AI model. Similarly, presence or absence of utilization values in all class encodings represented by an AI model (i.e., AI model fingerprint) allowed us to focus on specific subgraphs and hence reduce the search space for explaining differences between clean and poisoned classes. Finally, we documented several cost versus explainability tradeoffs in terms of computational requirements. As there is a race between explainability costs and the growth of training datasets and AI model sizes [42], we found the path toward explainability only if training images, inferred images, and trained AI models are available for in-depth analyses. If other than known training and inferred images are presented, then trust in predictions would be limited due to a tug of war between trojan detectors and trojan detectability (see Discussion section).

In the future work, we will analyze existing AI models that have been identified to contain natural triggers. We also plan to document the characteristics of physically realizable triggers found in traffic sign classifications models. Our future work is motivated by delivering fully trusted AI models for life-critical applications. In the context of our presented work, fully trusted AI models imply access to all tensor-states (activation maps) at each graph node and for each training image, as well as their overlaps with tensor-states invoked by other predicted classes (i.e., tensor-states common to two and more classes). Furthermore, we plan to contribute to symbolic representations of graphs and subgraphs in the context of training images consisting of semantically meaningful objects.

## CRediT authorship contribution statement

Peter Bajcsy: Conceptualization, Theory, Methodology, Software, Experiments, Data analyses, Writing - original draft preparation; Antonio Cardone: Data analyses - hypothesis testing, Writing - reviewing; Chenyi Ling: Data analyses - hypothesis testing; Philippe Dessauw: Data - AI model training, Writing - reviewing; Michael Majurski: Data - AI model training, Writing - reviewing; Tim Blattner: Data - AI model training; Derek Juba: Hardware - AI model training; Walid Keyrouz: Writing - reviewing and editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgement

The funding for all authors was provided by the Intelligence Advanced Research Projects Activity (IARPA): IARPA-20001-D2020-2007180011. We would like to acknowledge the contributions of Mylene Simon and Ivy Liang to develop an interactive web application for online comparison of AI model fingerprints. We would also like acknowledge Peter Fontana and Joe Chalfoun from National Institute of Standards and Technology for providing additional comments on the manuscript.

## Disclaimer

Commercial products are identified in this document in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the products identified are necessarily the best available for the purpose.

## References

- [1] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A. Gunter, and Bo Li. Detecting AI Trojans Using Meta Neural Analysis. <http://arxiv.org/abs/1910.03137>, 2019.
- [2] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.
- [3] Hima Lakkaraju, Julius Adebayo, and Sameer Singh. *Explaining Machine Learning Predictions: State-of-the-Art, Challenges, and Opportunities*. NeurIPS 202 Tutorial, 2020.
- [4] Chris Ying, Aaron Klein, Eric Christiansen, Esteban Real, Kevin Murphy, and Frank Hutter. NAS-bench-101: Towards reproducible neural architecture search. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*,

volume 97 of *Proceedings of Machine Learning Research*, pages 7105–7114, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

- [5] OpenAI. OpenAI Microscope, 2022. <https://microscope.openai.com/about>.
- [6] Daniel Justus, John Brennan, Stephen Bonner, and Andrew Stephen McGough. Predicting the Computational Cost of Deep Learning Models. <https://arxiv.org/abs/1811.11880>, 2019.
- [7] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In *NDSS*, pages 6232–6240, Long Beach, CA, 2017. Internet Society, Advances in Neural Information Processing Systems.
- [8] Nicholas J. Schaub and Nathan Hotaling. Assessing intelligence in artificial neural networks. <https://arxiv.org/abs/2006.02909>, 2020.
- [9] Andrea Bontempelli, Fausto Giunchiglia, Andrea Passerini, and Stefano Teso. Toward a unified framework for debugging concept-based models. <https://arxiv.org/abs/2109.11160>, 2021.
- [10] Asifullah Khan, Anabia Sohail, Umme Zahoor, and Aqsa Saeed Qureshi. A Survey of the Recent Architectures of Deep Convolutional Neural Networks. <https://arxiv.org/abs/1901.06032>, 2020.
- [11] Andrew Trask, David Gilmore, and Matthew Russell. Modeling order in neural word embeddings at scale. *32nd International Conference on Machine Learning, ICML 2015*, 3:2256–2265, 2015.
- [12] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *CoRR*, abs/2101.03961, 2021.
- [13] Intelligence Advanced Research Projects Agency IARPA. Trojans in Artificial Intelligence (TrojAI). <https://pages.nist.gov/trojai/>, January 2020.
- [14] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11050 LNCS:273–294, 2018.
- [15] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. <https://distill.pub/2020/circuits/zoom-in>.
- [16] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. <https://distill.pub/2017/feature-visualization>.

- [17] Peter Bajcsy, Nicholas J. Schaub, and Michael Majurski. Designing trojan detectors in neural networks using interactive simulations. *Applied Sciences*, 11(4), 2021.
- [18] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. <https://arxiv.org/abs/1806.00069>, 2018.
- [19] Richard J. Haier, Benjamin V. Siegel, C. Tang, Lennart Abel, and Monte S. Buchsbaum. Intelligence and changes in regional cerebral glucose metabolic rate following learning. *Intelligence*, 16:415–426, 1992.
- [20] Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. <https://arxiv.org/abs/2207.13243>, 2022.
- [21] Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. <https://arxiv.org/abs/1801.03454>, 2018.
- [22] David Bau, Zhou Bolei, Khosla Aditya, Oliva Aude, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proc. 2017 IEEE Conf. Comput. Vision and Pattern Recognition*, pages 3319—3327, 2017. <https://doi.org/10.1109/CVPR.2017.354>.
- [23] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, dec 2020.
- [24] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, oct 2019.
- [25] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *In Advances in Neural Information Processing Systems 31*, page 9505–15. Curran Associates, Inc., 2018.
- [26] Chihiro Watanabe, Kaoru Hiramatsu, and Kunio Kashino. Understanding community structure in layered neural networks. <https://arxiv.org/abs/1804.04778>, 2018.
- [27] Shlomi Hod, Daniel Filan, Stephen Casper, Andrew Critch, and Stuart Russell. Quantifying local specialization in deep neural networks. <https://arxiv.org/abs/2110.08058>, 2021.
- [28] Daniel Filan, Stephen Casper, Shlomi Hod, Cody Wild, Andrew Critch, and Stuart Russell. Clusterability in neural networks. <https://arxiv.org/abs/2103.03386>, 2021.

- [29] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. <https://arxiv.org/abs/1811.12359>, 2018.
- [30] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. A sober look at the unsupervised learning of disentangled representations and their evaluation. 21:1–62, 2020.
- [31] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [32] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [33] Joseph Bruno, Joseph Ingo, and Frese Daniel. Pexels: Photo and Video Sharing . <https://www.pexels.com/>, 2022.
- [34] Yuichi Asahiro, Hiroshi Eto, Takehiro Ito, and Eiji Miyano. Complexity of finding maximum regular induced subgraphs with prescribed degree. *Theoretical Computer Science*, 550:21–35, 2014.
- [35] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. page 1485–1488. Proceedings of the 18th ACM international conference on Multimedia, October 2010.
- [36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [37] Yu Zhang, Xiaofei Liao, Hai Jin, Bingsheng He, Haikun Liu, and Lin Gu. Digraph: An efficient path-based iterative directed graph processing system on multiple gpus. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS ’19, page 601–614, New York, NY, USA, 2019. Association for Computing Machinery.

- [38] Michael Majurski, Timothe Blattner, and Derek Juba. TrojAI code used to construct each round of the TrojAI challenge . <https://github.com/usnistgov/trojai-round-generation/tree/round4>, 2022.
- [39] Babak Hassibi and David G. Stork. Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. In *Advances in Neural Information Processing Systems 5 (NIPS 1992)*, pages 164–172. Information Processing Systems Foundation, Inc., 1992.
- [40] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning Filters for Efficient ConvNets. In *International Conference on Learning Representations*, pages 1–13, Palais des Congrès Neptune, Toulon, France, 2017.
- [41] Peter Bajcsy and Michael Majurski. Baseline pruning-based approach to trojan detection in neural networks. <https://arxiv.org/abs/2101.12016>, 2021.
- [42] Advait Sarkar. Is explainable AI a race against model complexity? <https://arxiv.org/abs/2205.10119>, 2022.
- [43] Si Liu, Risheek Garrepalli, Dan Hendrycks, Alan Fern, Debashis Mondal, and Thomas G. Dietterich. PAC guarantees and effective algorithms for detecting novel categories. *Journal of Machine Learning Research*, 23(44):1–47, 2022.
- [44] Mingjie Sun, Siddhant Agarwal, and J. Zico Kolter. Poisoned classifiers are not only backdoored, they are fundamentally broken. <https://arxiv.org/abs/2010.09080>, 2020.
- [45] Shafi Goldwasser, Michael P. Kim, Vinod Vaikuntanathan, and Or Zamir. Planting undetectable backdoors in machine learning models. <https://arxiv.org/abs/2204.06974>, 2022.
- [46] NIST. Documentation of TrojAI Challenge. <https://pages.nist.gov/trojai/>, 2022.
- [47] Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of National Academy of Sciences, USA*, 8(79):2554–8, 1982.
- [48] Y. Abu-Mostafa and J. St. Jacques. Information capacity of the hopfield model. *IEEE Transactions on Information Theory*, 31(4):461–464, 1985.
- [49] Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. <https://arxiv.org/abs/1606.01164>, 2016.
- [50] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Thomas Adler, David Kreil, Michael K Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp

Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021.

- [51] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.
- [52] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(623-656):379–423, 1948.
- [53] Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, oct 1989.
- [54] Peter L. Bartlett, Nick Harvey, Chris Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. <https://arxiv.org/abs/1703.02930>, 2017.
- [55] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, USA, 1st edition, 2009.
- [56] S. Kullback and R. A. Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics.*, 22(1):79–88, 2017.
- [57] Nicholas J. Schaub. Tensorstate toolbox. <https://tensorstate.readthedocs.io/en/latest/>, 2020.

## A Theoretical Reasoning and Relationships to Past Work

Binary tensor-states in AI computation graphs: While one computation unit in an AI computation graph produces real-value numbers, they can be binarized by thresholding at zero during measurement extraction. The thresholding operation reduces the real-value variability to  $\{0, 1\}$  by discretizing the output real values. One can model the discretization step according to Hopfield Neural Networks (HNN) [47] where a computation unit is viewed as an associative content-addressable memory for storing information bits about training data points [48]. The HNN model was motivated by collective properties of neurons in neurobiology adapted to integrated circuits [47], and the more recent work on dense associative memory [49] and Hopfield layers [50]. The thresholding value at zero is motivated by zero-centered normalizations of input data and the majority of nonlinear activation functions being centered around zero (i.e., tanh, sigmoid, rectified linear unit (ReLU), or leaky ReLU).

Utilization With Respect to State Representation Power of a Computation Unit: A particular computation unit in an AI computation graph can represent a maximum number of states depending on the number of outputs. For example, a fully connected layer (one computation unit) consisting of two nodes (two scalar outputs) can represent no more than four states, such as  $\{00, 01, 10, 11\}$ .

This fully connected layer would be fully utilized if all four states were used for predicting output labels.

In order to define the utilization of a computation unit mathematically, one can use the parallels drawn by Bajcsy et al. [17] between neural network and communication fields in terms of (a) computation unit maximum representation power and channel capacity in communications and (b) computation unit utilization and channel efficiency while leveraging the universal approximation theorem [51] and the source coding theorem [52]. Using the theorems, a computation unit  $v_j$  with  $D_j^{Out}$  nodes has the maximum representation power (or computation unit capacity) of  $n_j = 2^{D_j^{Out}}$  possible states.

The utilization definition can also be related to Vapnik-Chervonenkis (VC) dimension of a class of binary classifiers  $H$  from the image space  $\chi$  to the label space  $\{0, 1\}$ ;  $H : \chi \rightarrow \{0, 1\}$  [53], [54]. The VC dimension  $VCdim(H)$  is defined as the size  $m$  of the largest shattered set of inputs  $\{\vec{x}_1, \dots, \vec{x}_m\}$  such that the growth function  $\Pi_H(m) = \max_{\vec{x}_1, \dots, \vec{x}_m \in \chi} |\{(h(\vec{x}_1), \dots, h(\vec{x}_m)) : h \in H\}| = 2^m$  [54]. For real-valued functions  $\mathcal{F}$  present in neural networks, one can define  $VCdim(\mathcal{F}) = VCdim(\{sgn(f) : f \in \mathcal{F}\})$ , which is achieved by thresholding at zero in the utilization computation. The pseudodimension of  $\mathcal{F}$  or  $Pdim(\mathcal{F})$  has been proven to satisfy  $VCdim(\mathcal{F}) \leq Pdim(\mathcal{F})$  [55]. For example, a convolutional computation unit  $v_j$  with the output tensor (*Channels*, *Width*, *Height*) per input image  $\vec{x}_i$  would yield  $VCdim(\mathcal{F}(v_j)) = Channels$  after binarizing the output values. The  $VCdim$  values are used to quantify utilization of each computation unit as documented in Section 3. Our work does not currently leverage the possibility of using lower and upper bounds on  $VCdim(\mathcal{F})$  for an entire neural network. These bounds have been derived for AI architectures with computation units characterized by constant and known non-linearities (piece-wise constant, piece-wise linear, and piece-wise polynomial [54]) with respect to the number of layers  $L$  and parameters  $W$ . While in practice there is no guarantee that computation units in one architecture are not characterized by a mixture of non-linearity types, it would be possible to verify the theoretical bounds for a combination of  $L$  and  $W$  in a variety of existing popular AI architectures.

Analytical and Statistical Utilization Definitions: Given a set of training data points activating a computation unit in an AI computation graph, the computation unit will generate unique output states at varying frequencies. Considering that training datasets are formed by sampling and split into train, validation, and test subsets, the utilization value would vary depending on the sampling techniques and subset of samples. Thus, one can define the utilization  $\eta_j$  of a computation unit  $v_j$  using analytical or statistical views since the states generated as outputs of a computation unit by evaluating a set of training data points can be interpreted as deterministic states or as a statistical distribution of states.

An analytical definition of computation unit utilization  $\eta^{state}$  is directly derived as a ratio of a number of unique tensor-states and a computation unit capacity. One could aim at optimizing a network architecture search (NAS)

[4] such that every computation unit reaches maximum utilization  $\eta^{state} = 1.0$  over all training data points and all predicted labels. A statistical definition of computation unit utilization is derived from a tensor-state distribution shape and aims at maximum normalized entropy of the state distribution for each predicted label (entropy-based utilization)  $\eta^{entropy}$  or at minimum Kullback–Leibler (KL) divergence of the state distribution from a uniform distribution allocated per predicted label  $\eta^{KLDiv}$ . The use of Entropy and KL divergence [56] are borrowed from the source coding theorem [52].

Ordering Utilization Values in Class Encodings and AI Model Fingerprints: For image classification AI models, a training dataset consists of one set of input images per output class label. Performing an inference on a set of training images per class label will produce one of the three utilization values per measurement probe (e.g., right after each computation unit of a computation graph). The order of probes can be determined based on the flow of input images through a computation graph since it is deterministic and well-defined during an inference computation. Given a sorted (ordered) list of probes and their utilization values, *a utilization-based class encoding* is uniquely defined.

An AI model predicts multiple class labels, and, therefore, utilization evaluations of the same AI model will consist of multiple class encodings. Although class labels have semantic annotations (i.e., cat or dog), they are typically numerically labeled as well, and, therefore, they can be ordered. Given a sorted (ordered) list of classes by numerical labels, *a utilization-based fingerprint of an AI model* is well-defined as a matrix of ordered utilization-based class encodings.

## B Verification of Utilization Measurement Properties

Utilization measurements must satisfy a few expected properties listed below:

1. *Inferred data:* Average utilization must be nondecreasing with increasing number of data points used for utilization measurements - see Figure 29.
2. *Predicted classes:* Average utilization must be nondecreasing with increasing number of predicted classes - see Figure 25.
3. *AI model capacity:* Average utilization must be nondecreasing with decreasing capacity of an AI model - see Figure 26.

Since the KL Divergence-based utilization measures ‘non-utilization’ (or inefficiency), the trends demonstrating the properties are reversed. The low and high utilization values are highlighted along a vertical axis on the right side of Figures 25, 26, and 29.

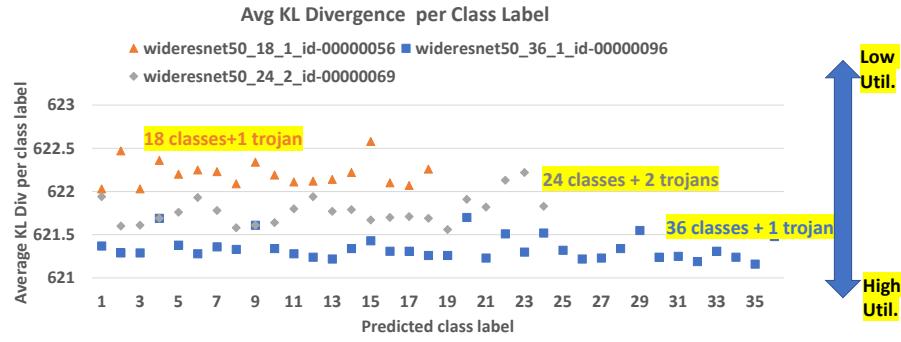


Figure 25: Average KL Divergence utilization over all probes as a function of the number of predicted classes in WideResNet50 architecture.

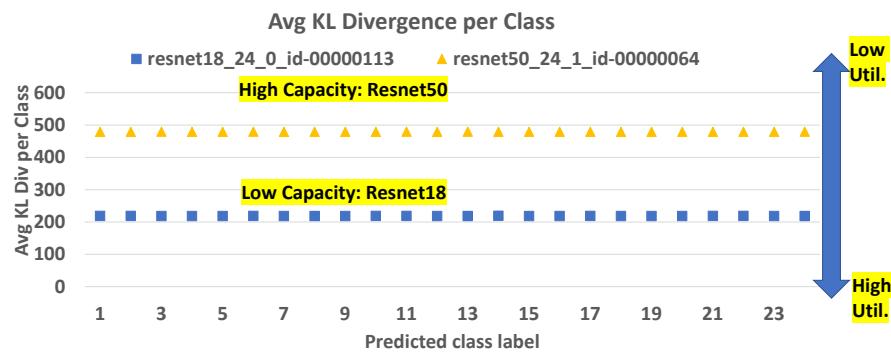


Figure 26: Average KL Divergence utilization over all probes as a function of ResNet AI model architectures. ResNet18 and ResNet50 differ by including 20 or 50 convolutional layers affecting their modeling capacity.

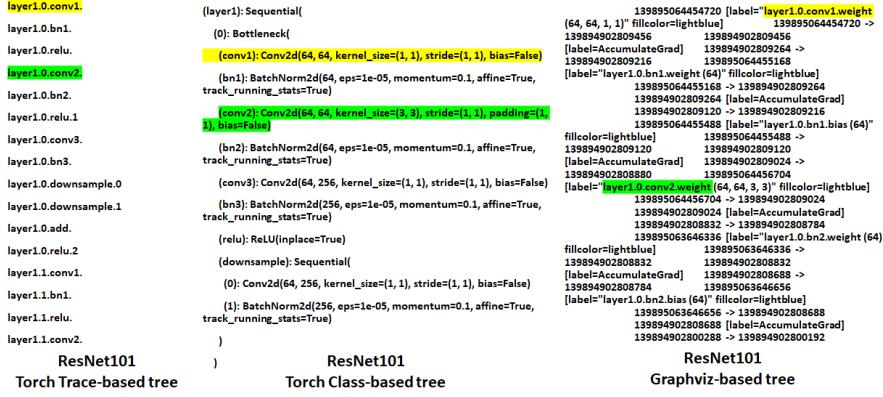


Figure 27: ResNet101 graph representation (snippets) extracted by using (1) Torchvision library (left: trace-based tree, middle: block class-based tree) and (2) GraphViz library (right: nodes and edges). The highlighted text shows corresponding entities in the three tree representations.

## C Identification and Placement of Utilization Measurement Probes

Every utilization measurement (every probe) has its associated cost in terms of computer memory and execution time. The placement of measurement probes can be optimized so that the number of probes is minimal and the explainability is maximal. The challenges in optimal placement come from the fact that AI model computation graphs are not only structurally very complex but also very heterogeneous in terms of computation units.

To automate the utilization measurements, we identified computational units in all 35 image classification architectures supported by Torchvision [35]. The classification architectures of AI models are extracted into trace-based tree or block class-based tree textual representations - see Figure 27 (left and middle). Using the TorchScript vision library [35], these tree representations can be converted into a DiGraph representation [37] (see Figure 27 right) that can be pseudo-colored and visualized according to the block diagram shown in Figure 13.

To simplify the automated placement of measurement probes (i.e., software hooks), we extracted a list of unique computation units in block class-based trees of all Torchvision supported classification architectures and used the TensorState library [57] for placing the measurement probes within each occurrence of a class. TensorState attaches measurement probes after each computation unit within a class and collects tensor-state statistics during image inferences. The placement of measurement probes after every computation unit yields a baseline for studying cost versus information tradeoffs.

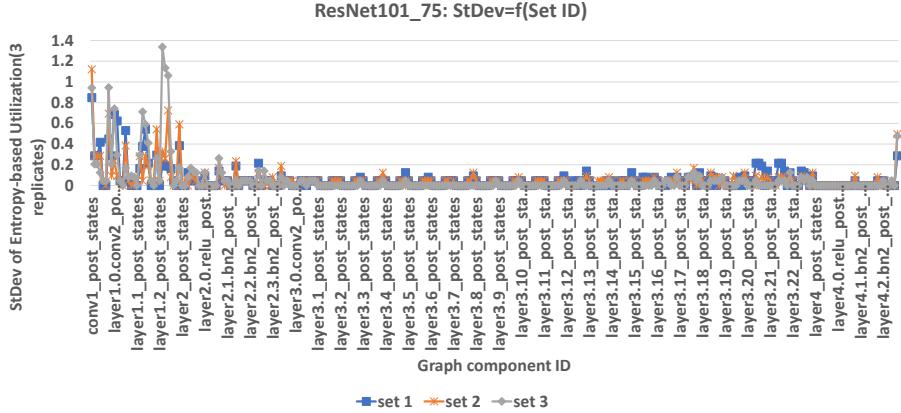


Figure 28: Standard deviation of Entropy-based utilization at each graph location (probe) using the three sets of images defined in Table 2.

## D Variability of Class Encodings in computation graph

Each trained AI model is unique in the way it encodes classes in computational units of computation graphs. We quantified the class encoding variability by averaging utilization values from three AI models trained on the same training dataset with varying random master seed for each training session and computing its standard deviation. The average utilization values varied between 0.5 % and 30 % with higher values in graph computation units before reaching layer3.

Figure 28 shows the standard deviation of utilization values per computational unit for three clean and three poisoned ResNet101 models predicting 75 traffic signs. The clean models were evaluated using clean images (Set 1) and resulted in three utilization values per graph component. The poisoned models were evaluated with clean images (Set 2) and poisoned images (Set 3) as defined in Table 2 and also resulted in three values per graph component per Set. The models were poisoned with Kelvin Instagram filter, which is boosting the earth tones of green, brown, and orange. The largest standard deviation was 1.34 % measured in layer1.2 by evaluating poisoned AI model on Set 3. The second largest value was 1.11 % measured in the first conv1 computation unit by evaluating poisoned AI model on Set 2. The larger magnitudes of utilization standard deviation in poisoned AI models over clean AI models suggest a higher variability in how trojans are encoded during each training session.

## E Reduction of Computational Requirements

Following up on the computational complexity of utilization measurements in Section 3.2, we gathered utilization values for varying number of images per class

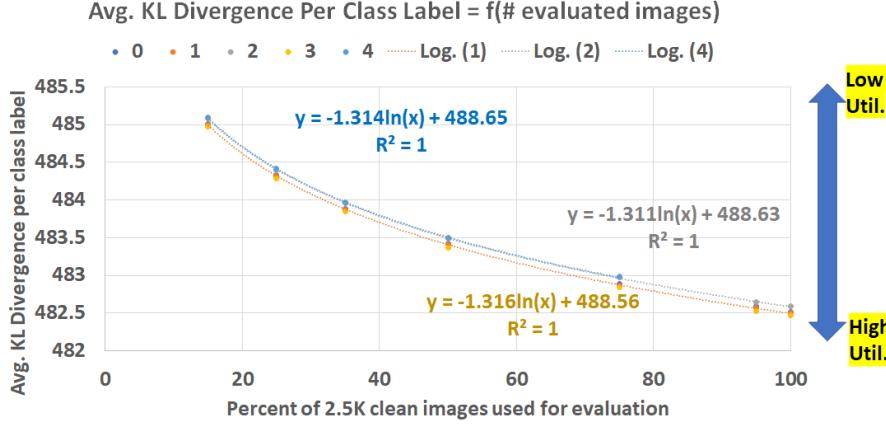


Figure 29: Average KL Divergence utilization over all probes per class label as a function of reduced number of training images. The utilizations are evaluated on subsets between 15 % and 100 % of the 2.5K entire image collection for classes 0, 1, 2, 3, and 4. Three of the utilization curves are approximated with a  $\ln$  function.

in order to build an extrapolation model and save some computational time. Figure 29 shows average KL divergence-based utilization as a function of a reduced number of images per class. The extrapolation model is consistent across multiple predicted classes with average parameters  $\eta_j^{KLDiv} = -1.314\ln(M) + 488.61$  where  $M$  is the number of training images and the R-squared value equal to one ( $R^2 = 1$  implies high trendline reliability).

The execution time as a function of the number of evaluated images is approximated with a linear model: Time [s] =  $242.75 * M - 497.02$  ( $R^2 = 0.993$ ) where  $M$  is the number of input images sampled at  $\{15, 25, 35, 50, 95, 100\}$  percent of 2500 images per class. In our specific case, the time reduction was from 6.5 h for 100 % to 0.9 h for 15 % of evaluated images per class for a ResNet101 model predicting 40 traffic sign classes.

We also documented a computational tradeoff in ResNet101 between the number of measurement probes and the average computational time in Table 7. Based on the numerical values, the functional model is non-linear since the execution time depends not only on the number of probes but also on the complexity of the state tensors generated by each graph node.

Table 7: Computation time of utilization as a function of the number of measurement probes and placement in the ResNet101 architecture (sorted by time).

<b>Probe name</b>	<b>Number of probes</b>	<b>Avg. Time [s]</b>
“layer3” Block	1	80.67
“conv2” Node in all 4 “layer” Blocks	4	87.95
Layer1.Bottleneck0 Block	1	88.03
Fully Connected Block	1	106.53
Top “conv1” Node	1	107.65
All Nodes “ReLU”	34	258.95
All in “Conv2d” Block	104	280.26
All Nodes	286	868.19

# AI and the FCI: Can ChatGPT project an understanding of introductory physics?

Colin G. West<sup>1\*</sup>

<sup>1</sup>Department of Physics, University of Colorado, Boulder, Colorado 80309, USA

(Dated: March 3, 2023)

ChatGPT is a groundbreaking “chatbot”—an AI interface built on a large language model that was trained on an enormous corpus of human text to emulate human conversation. Beyond its ability to shoot the breeze in a plausible way, it has attracted attention for its ability to competently answer questions from the bar exam and from MBA coursework, and to provide useful assistance in writing computer code. These apparent abilities have prompted discussion of ChatGPT as both a threat to the integrity of higher education and conversely as a powerful teaching tool. In this work we present a preliminary analysis of how ChatGPT fares in the field of first-semester university physics, using primarily the Force Concept Inventory (FCI) to assess whether it can give correct responses to conceptual physics questions about kinematics and Newtonian dynamics. We demonstrate that, by some measures, ChatGPT can match or exceed the median performance of a university student who has completed one semester of college physics, though its performance is notably uneven and the results are nuanced. We conclude with a discussion of these results in light of four questions that motivated this work: what does ChatGPT’s performance tell us about the nature of conceptual assessment tools like the FCI? How might the availability of ChatGPT as a resource for students? Can ChatGPT be used as an in-class teaching tool for physics instruction? And can it be used as an out-of-classroom aid to those engaged in physics pedagogy?

## I. INTRODUCTION

“ChatGPT,” in simplest terms, is a software application designed to mimic human conversation by producing and responding to text, a skill called “natural language processing.” [1] Technically, it is based on a “large language model” (LLM) called “ChatGPT3.5” which makes use of two recent advances in the LLM field: the “Transformer” model [2] and “pretraining.” [3] from whence arises “GPT” (it is a [G]enerative, [P]retrained [T]ransformer model). While a great deal has been written about the methods used to produce the ChatGPT system [4], it suffices here to note that it is one of a new generation of artificial language processing systems—sometimes colloquially called “chatbots”—which has garnered substantial attention in both academic [5] and popular press [6] for its ability to seemingly carry on a coherent conversation and complete other tasks.

In addition, recent papers have shown that, by at least some measures, ChatGPT’s ability to converse like a human also allows it to seemingly display competence in fields like business and law. For example, its responses to questions about Operations Management, a core topic in many MBA programs, were assessed in one study as being at the “B or B-” level. [7] Another work concluded that, “although ChatGPT would have been a mediocre law student, its performance was sufficient to successfully earn a JD degree from a highly selective law school” [8]. A similar work projected that, given the surprisingly strong performance of ChatGPT on sample questions it could not have seen before, a similar LLM might be able to pass an actual bar exam “within the next 0-18 months.” [9] [10] Not knowing much about business and

law, we ask a similar question in our own area of expertise and explore how ChatGPT’s responses to questions about introductory physics compare to those of physics students and of expert physicists.

Beyond the obvious motive of innate curiosity and a shameless desire to participate in the buzz surrounding ChatGPT, we identify four major motivations for this work. First, an exploration of how an automated system performs on a standardized assessment of conceptual physics can give us an improved sense of what our assessment tools do and do not measure. Second, it provides important insight into a kind of tool that our students might soon be using for help with online assignments, written homework, and more, which is already happening in other fields of study [11]. Third, it illuminates some of the potential (and limitations) of ChatGPT as an in-class tool, which has also begun in various other fields of higher education [12]. Finally, it provides some initial ideas for ways in which physics educators could use ChatGPT and similar tools in their own work and preparations for teaching.

## II. BACKGROUND

It is important to remember that ChatGPT was designed and optimized *specifically* for the art of conversing in a manner that would seem plausibly human [4]. To a loose approximation, it has analyzed and internalized the patterns of words in an enormous sample of human text (principally the “CommonCrawl” dataset) [4]), which consists of things like books, news articles, wikipedia pages, reddit threads, and content from more specialized and technical fora like StackExchange and StackOverflow [4, 13]. From this data, it has an immense probabilistic model of how words tend to be fit together by a

---

\* colin.west@colorado.edu

human being in various context. For example, it "knows" that a sentence which begins "after the fight, William hurt..." is likely to end with something like "...his hand." But it also recognizes that the sentence "after the fight *scene*, William Hurt..." might instead end with "... appeared to be injured and had to be helped off set." [14]. Crucially, this means that ChatGPT is a tool which manipulates and responds to language, and is not designed or trained to implement any model of an underlying concept [15] like business, law, or physics. But it is designed to talk like a person, and people who talk about business, law, and physics generally discuss these topics with some intelligence (at least, relatively speaking). Hence, even without any specific training in physics, it's familiarity with the way physicists talk about physics may be enough for it to project at least an appearance of understanding.

The concept of "Understanding" can be deep and difficult to define, particularly in the context of learning a new topic [16]. Hence, while we have our own opinions about whether ChatGPT's behavior actually constitutes an act of "understanding" the topics it can discuss [17], we will generally leave that longstanding question [18] to the likes of AI researchers, cognitive scientists and philosophers. For this reason we have been very careful with the wording in the title of this paper and elsewhere: we are not about to ask, because we are not equipped to know, whether ChatGPT *understands* introductory physics. We ask whether its behavior *creates the appearance of understanding* to the outside world— which might either be because it has succeeded in understanding, or merely because it has figured out how to display all the usual indicia of understanding. Hence, in the same manner that some poker player might "project confidence" either as a proud display of internal fortitude or as an act desperate bluffing, we consider here whether ChatGPT "projects" an understanding of introductory physics to the outside world.

There is a deep body of work in the literature of physics education about how to assess whether a student is demonstrating "understanding," [16, 19]. To start with the basics, we choose as our primary assessment tool a classic instrument which has been used almost as a gold standard for decades: the Force Concept Inventory (FCI) [20]. This influential and heavily-studied assessment is a set of 30 questions designed to try to isolate and allow students to demonstrate a *conceptual* understanding of introductory kinematics and dynamics, as they might be covered in the first semester of introductory physics at the high school or university level. Although it has plenty of limitations, it was designed as a tool with an eye to distinguishing true conceptual mastery from the kinds of rote memorization, pattern-matching, and algorithmic calculation[21–23] which students sometimes use in order to pass conventional physics tests without ever truly knowing what they are doing, or why. Its status as a classic test for understanding in introductory physics makes it a logical starting point for our inquiry, though

we encourage future work to extend beyond this starting point as well.

In short, we will offer ChatGPT a modified version of the questions from the FCI, and assess on multiple levels how successfully it can project understanding of these topics in intro physics. Section III of this paper we describe in more detail how the assessment was modified and administered, and in section IV we analyze ChatGPT's responses through various lenses and compare its performance to a large sample of real human students. Section V gives a summary of our resulting inferences and speculations, as well as a discussion of directions for future work.

### III. METHODS

Before presenting our results, we will discuss in more detail our reasons for using the FCI, the steps taken to convert it to a form that could be administered to ChatGPT, and the procedures followed during its administration. Because all of our reference points for what it looks like when someone projects an understanding of physics are based on the performance of other human beings, Hence, it was important when administering the FCI that we hew as closely as possible to the conditions as our human students would encounter it. This was not entirely possible, and some modifications were necessary both to the questions themselves as well as to the process of giving them to ChatGPT. But such changes were kept to a minimum. In this section we detail those changes and the resulting procedure for assessing ChatGPT with the FCI.

#### A. Suitability of the FCI

The FCI is a natural fit as a first step in assessing the capabilities of ChatGPT for several reasons mentioned above: it is focused on conceptual understanding rather than computation and memorization; it has been widely-studied and validated as an assessment tool, and because it has been given frequently to many students in introductory physics, providing natural benchmarks for comparison. But it is also potentially valuable for another reason. To preserve its integrity as an assessment tool, the providers of the FCI have taken steps to encourage practitioners who use it to keep its contents (and even more importantly, its solutions) from becoming widely available. These efforts have certainly not been flawlessly successful but it remains the case that FCI text and solutions are difficult to find on the internet. Where they do exist, they are typically either password-protected on websites used by physics educators or at least "paywalled" on websites used to undermine physics educators, whose profit motive for getting students to pay for easy answers to problems conveniently also helps to minimize their availability. The relative scarcity of

FCI solutions on the internet (certainly when compared with, say, the end-of-chapter problems from popular [24] introductory textbooks) is consistent with the findings of prior work showing that access to the internet does not undermine the validity of tools like the FCI as conceptual assessment, even when students are observed to be copying question text for the presumable purpose of searching for its answers [25]. And in our case, since we are administering the FCI to a program that can only parse text, we can go even further: a significant portion of the FCI problems and/or solutions that can be found online exist as scanned images and/or had-annotated PDFs, which means that they would not be parsed by the kinds of automated tools that scrape the web for text.

While the exact details of the text which was used to “train” ChatGPT are a proprietary secret, it is known that its reading material was largely drawn from the “Common Crawl” corpus [13], an open repository of data scraped from text found on the public internet. Common Crawl allows users to query which domains it has indexed; we used this feature to verify that it has not indexed the handful of websites which we are most familiar with which might contain solutions to the FCI, either as a PER tool or as a repository of solved problems for students. Beyond the CommonCrawl corpus, it is believed that most of ChatGPT’s training data came from specifically generated human with human feedback [26], which of course are highly unlikely to contain references to the FCI outside of perhaps the small bubble which the author and his colleagues inhabit. And finally, in the rare locations that we were able to find FCI solutions online, they were typically stored apart from the questions themselves, meaning that there was no obvious reason an LLM would know to pair particular solution texts with particular problems even if it had access to them. For all these reasons, we believe that the FCI is likely *not* in the training text of ChatGPT and hence that its responses have to represent more than regurgitation of something it “remembers.” In this respect our testing with the FCI is what researchers in AI and machine learning might term “zero shot task”: a challenge in which the model is used to classify (and in this case, respond to) prompting text it has never seen before.

## B. Modifying the FCI

The FCI is a 30-item sequential multiple-choice assessment, with each item containing five choices (four distractors and one unique correct answer or “key”). [20] It’s items cover a range of topics from approximately the first third of a semester of college-level introductory physics: kinematics, projectile motion, free-fall, circular motion, and Newton’s laws. This means that generally, it is well-suited to our task. It’s one significant drawback is that 18 of its 30 items contain some kind of reference to a figure. ChatGPT is designed only to accept text input, and despite some clever attempts to feed it images in some

sort of indirect or transformed state [27], it does not seem capable currently of extracting any meaningful information from a picture, let alone analyzing it with the level of detail needed to answer a physics question.

Of the 18 items with figures, 11 of them could be modified by adding text that described what was shown in the figures without fundamentally altering the task at hand. In doing so, we took care to make sure that we did not provide additional clues or context that would make the problem simpler for ChatGPT than it would be for a typical physics student. For example, item seven involves a steel ball on a rope being swung in a circular path and then suddenly cut free. The question asks about the path of the ball after it is released, and the figure supplies several different possible trajectories. One way to describe these trajectories in words would be to say “tangential to the circle,” “normal to the circle,” etc. But we suspect this modification would substantially alter the difficulty of the problem. After all, the FCI is meant to test a *conceptual* understanding of physics, and a human student might be able to answer a question that had been modified in this way simply by rote recollection that the word “tangential” seemed to come up a lot in reference to objects in circular motion, even without knowing what that meant or why it was relevant. ChatGPT could easily draw the same inference as a matter of linguistic association, without displaying anything resembling evidence of understanding. Instead, we translate this figure into words with reference to cardinal directions:

Consider a moment in the ball’s motion when the ball is moving north. At that moment, the string breaks near the ball. Which of the following paths would the ball most closely follow after the string breaks?

- (a) It will initially travel north, but will quickly begin to curve to the west
- (b) It will travel north in a straight line
- (c) It will travel northeast in a straight line
- (d) It will initially travel east, but will quickly begin to curve north
- (e) It will travel East in a straight line

We feel that this wording captures exactly and unambiguously all of the different paths indicated in the original figure, but without providing any additional hints. If anything, it may make the item slightly harder for ChatGPT than the originals.

Thirteen other items from the FCI were modified in a similar way. Six items without figures received minor text modifications that should not have affected the nature of the physics being tested. For example, in clusters of questions where some items referenced “the previous problem,” we removed these references and simply restated the set-up from the prior problem, so that items could be asked about independently if needed. We

also rephrased any questions that were left with an open-ended statement for the student to complete, since initial experiments showed that ChatGPT occasionally appeared “confused” when it was not explicitly asked a question. Hence, a question like item one, which originally read:

Two metal balls are the same size but one weighs twice as much as the other. The balls are dropped from the roof of a single story building at the same instant of time. The time it takes the balls to reach the ground below will be:

- (a) About half as long for the heavier ball as for the lighter one
- (b) About half as long for the lighter ball as for the heavier one
- (c) About the same for both balls
- (d) Considerably less for the heavier ball, but not necessarily half as long
- (e) Considerably less for the lighter ball, but not necessarily half as long

Was rephrased to end with a direct question:

Two metal balls are the same size but one weighs twice as much as the other. The balls are dropped from the roof of a single story building at the same instant of time. Which of the following describes the time it takes the balls to reach the ground below?

- (a) About half as long for the heavier ball as for the lighter one
- (b) About half as long for the lighter ball as for the heavier one
- (c) About the same for both balls
- (d) Considerably less for the heavier ball, but not necessarily half as long
- (e) Considerably less for the lighter ball, but not necessarily half as long

We do not imagine that such changes impacted either the physics content or the difficulty of the items. Finally, four of the items were left entirely unchanged.

This meant that we were able to ask ChatGPT 23 of the FCI’s 30 items. Although others have shown that it is possible to get a representative sample of a student’s performance using only a subset of the FCI questions [28], it happens that the “unusable” questions are not uniformly distributed across all question categories. Removing problems 19 and 20, for example, meant removing the only questions on linear kinematics from the instrument. Although this affects our ability to make comparisons with results from the “full” FCI, we believe this difficulty can be overcome, as we shall discuss in Sec IV below.

Type of change	Items
None	1, 4, 29, 30
Minor text	2, 3, 13, 25, 26, 27
Figure description	5, 7, 9, 10, 11, 15, 16, 17 18, 22, 23, 24, 28
Unusable	6, 8, 12, 14, 19, 20, 21

TABLE I: Table of items from the FCI and the ways that they were (or were not) modified for use in this work. Seven items were not used.

### C. Administering the FCI to ChatGPT

We began interacting with the version of ChatGPT which existed during the month of January, 2023, and used our initial explorations there to develop a the guidelines we used for how to pose questions. Instances of conversations with ChatGPT are completely separate, in the sense that ChatGPT does not “remember” content from one chat in a separate chat, so variations of a question can be asked in parallel to identify the best practices for posing the questions. On the other hand, *within* a conversation ChatGPT can remember content back to a depth of about 3000 words. But in practice, this is not enough to remember a full administration of the FCI. This is part of the reason that we chose to rephrase each question so that it could stand alone, rather than referencing things from “the previous question,” etc.

A prior work, in which bar exam questions were administered to ChatGPT, found various tricks that caused it to perform better with multiple-choice questions [9] (the art of tweaking the input to an LLM to optimize its response in this fashion is called “prompt engineering.”) In particular, they found that that, rather than asking ChatGPT for a single answer, it performed better when it was asked to rank its top three choices (though in actuality only it’s top choice was scored). It is the nature of an LLM that, much like a tantrum-throwing toddler still learning to talk, we cannot know exactly how or why it responds better to some feedback than others. The authors of Ref. [9] speculate quite convincingly that this approach “best combined non-entailment performance, i.e., rejection of most incorrect answer, with probabilistic entailment and recall for remaining choices.” We therefore administered the FCI to ChatGPT twice (recall that it cannot share memory between conversations), once using a more conventional multiple-choice prompt, which we call the “BASIC” prompt, and once using this rank-ordering approach, which we call the “RANKED” prompt.

Questions with the “BASIC” prompt were each posed like this:

Two metal balls are the same size but one weighs twice as much as the other. The balls are dropped from the roof of a single story building at the same instant of time. Which of the following describes the time it takes the

balls to reach the ground below?

- (a) About half as long for the heavier ball as for the lighter one
  - (b) About half as long for the lighter ball as for the heavier one
  - (c) About the same for both balls
  - (d) Considerably less for the heavier ball, but not necessarily half as long
  - (e) Considerably less for the lighter ball, but not necessarily half as long
- Please answer with a letter (A, B, C, D, or E) and a brief explanation of your reasoning.

Drawing directly from Ref. [9], questions with the “RANKED” prompt were posed like this:

Please answer the following physics question in the following rank order format: First Choice: `|LETTER|` Second Choice: `|LETTER|` Third Choice: `|LETTER|`

Here is the question:

Two metal balls are the same size but one weighs twice as much as the other. The balls are dropped from the roof of a single story building at the same instant of time. Which of the following describes the time it takes the balls to reach the ground below?

- (a) About half as long for the heavier ball as for the lighter one
- (b) About half as long for the lighter ball as for the heavier one
- (c) About the same for both balls
- (d) Considerably less for the heavier ball, but not necessarily half as long
- (e) Considerably less for the lighter ball, but not necessarily half as long

These two methods formed the basis for our primary exploration of ChatGPT’s ability to project understanding of introductory physics. Subsequently, we experimented with two additional procedures. First, we took advantage of feature offered by ChatGPT which allows the user to request that it “regenerate response” after it finishes its output. At a gross level, this feature is similar to asking an algorithm for numerically solving some equation to start again but with a different random initial guess. One expects that the results will generally converge to two similar outputs, but perhaps not arrive at exactly the same point. We used this feature as a rudimentary way to explore the “stability” of ChatGPT’s responses, which might in turn be thought of as a proxy for its “confidence” in its answers. For reasons discussed in our results section, we performed this stability analysis only with the “BASIC” style prompt. Our very preliminary

results based on this experimentation are discussed in Sec. IV A 3.

Finally, we also experimented with a very different prompt with a very different objective, which we call the “NOVICE” prompt. In this prompt each question was posed to ChatGPT in the following format:

Please answer the following question as though you were a novice high-school student who has not studied physics and does not yet understand Newton’s laws:

A large truck collides head-on with a small compact car. During the collision, which of the following is true:

- (a) the truck exerts a greater amount of force on the car than the car exerts on the truck.
- (b) the car exerts a greater amount of force on the truck than the truck exerts on the car.
- (c) neither exerts a force on the other, the car gets smashed simply because it gets in the way of the truck.
- (d) the truck exerts a force on the car but the car does not exert a force on the truck.
- (e) the truck exerts the same amount of force on the car as the car exerts on the truck.

Just give us your best guess. We know you may not know the correct answer, but we’d like to know which answer makes the most sense to you without any formal physics training.

We landed on this particular prompt after some trial-and error using the particular question above (item 4 on the FCI) for which we had a strong sense about what answer a novice student might choose and why. In particular, in a prior administration of the FCI in one of our introductory physics courses to a large sample ( $N = 415$ ) of students, 74% chose answer “A.” It took some work to find an otherwise-neutral prompt that would induce ChatGPT to choose this patently incorrect answer, but having found it, we administered the entire FCI one more time with this prompt in order to explore not only whether it could display an expert understanding of physics but also whether it could selectively display a “novice” understanding that realistically mimicked what we are used to from our students before they have formally studied physics. After all, one of our motivations for this work was to see whether ChatGPT could potentially be useful as a tool for instructors to be able to preview and probe the thinking of a sample “novice” student while they prepare their teaching materials. The results from this administration of the FCI with the “NOVICE” style of prompt are discussed in section IV C.

All four administrations of the FCI (with the different prompt types discussed above) were given during the weeks of Feb 13 and Feb 20, 2023. Notably, between our

initial experimentation in January and the final administrations of the FCI which generated the results below, there was a significant update to the model which focused on improving its mathematical capabilities, following a series of relatively high-profile examples where users were able to get ChatGPT to espouse manifestly untrue statements about elementary mathematics [29]. Since none of the items in the FCI involve calculations, we think it is unlikely that the update had much impact on our main results.

## IV. RESULTS

We begin with an analysis of ChatGPT's responses to the 23 usable FCI questions in the modified form described above. For the first, “BASIC” administration of these problems, we analyze the responses on both a quantitative level, focused purely on its multiple-choice response, and on a qualitative one, by analyzing its stated reason for the answer it chose. For subsequent variations we focused only on the quantitative multiple-choice response, because we did not observe significant variation in the free-response answers based on the prompt, though there is room here for a deeper dive into these in subsequent work.

### A. “BASIC” responses

#### 1. Answer Choices

When prompted with the 23 usable FCI questions using the “BASIC” prompt format, ChatGPT gave a correct answer for fifteen of them. It is not completely clear how we should interpret this number in order to make comparisons to human students taking the FCI. It is tempting to assign ChatGPT a “score” of  $15/23 \approx 65\%$ , but it is not entirely clear that this is fair. An argument could instead be made that its “score” is  $15/30 = 50\%$ , because of course one important aspect of “understanding” a physics problem is the ability to synthesize the data being presented across multiple representations [30]. Since ChatGPT simply cannot comprehend a question that requires reference to a figure, it could be said that it manifestly displays no understanding of that particular item.

In either case, however, ChatGPT’s quantitative performance compares quite adequately with the post-test results that are typical for students taking the FCI at the end of their first semester of college-level physics. We can make a direct comparison to one of the author’s previous classes in 2018, in which 415 students took the FCI at the end of the term, and produced the distribution of scores found in figure 1. In that distribution, the median student score was a 56%, meaning that depending on how charitably one treats its partial results, ChatGPT was either just below the performance of a typical

student or else perhaps a nontrivial cut above average. As an alternative perspective, the median letter grades of the students scoring around a 50% or a 65% were a B- and a B+, respectively, though this may speak as much to the author’s inability to combat grade inflation as to ChatGPT’s capabilities.

Of course, if one wishes to truly give the machine the benefit of the doubt, it should be recalled that ChatGPT has not had any formal physics training, so in a sense its proper comparator is the *pretest* scores of our introductory physics students. For the 2018 term referenced above, ChatGPT’s score of 15/30 would have put it at the 86th percentile; it’s more generously computed score of 15/23 would have put it at the 93rd percentile.

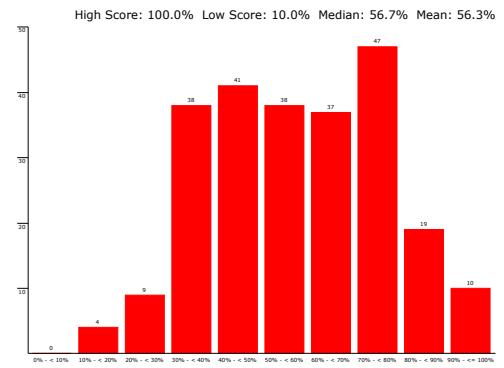


FIG. 1: The distribution of FCI scores at the end of a semester of college physics at a large public R1 university. ChatGPT’s performance using “BASIC” prompting would put it either in the bin containing the median (50%) or in the bin just above (65%)

#### 2. Free Responses

As part of the “BASIC” prompt, we asked ChatGPT to provide “a brief explanation of [its] reasoning.” This gives us an additional dimension along which to assess its performance, albeit a qualitative one.

As a baseline, we read through each response and made a decision about whether the answer was one that an expert physicist could plausibly have given. In this determination we disregarded tone and focused only on content. So, ChatGPT was not penalized if its correct explanation sounded stilted, in a way that suggested machine learning rather than a human intelligence behind the response. On the other hand, we marked “wrong” any response that contained an error which a trained physicist would not reasonably have made, no matter how ancillary the error was to ChatGPT’s analysis. Certainly, these determinations were simply a judgement call on the part of the author and of course were not blind to the fact that the responses were generated by a language model. However,

we feel that in most cases the determinations were relatively black-and-white, since they centered on a decision about whether the statements made in the explanation were factually true.

In our judgement, 10 of ChatGPT’s 23 free responses (43%) passed this “expert test.” These responses were typically relatively terse, invoking the correct constellation of concepts, and often making explicit note of facts or concepts which ruled out wrong answers as well as the facts which supported the correct one. For example, item 13 of the FCI says

“A boy throws a steel ball straight up. Consider the motion of the ball only after it has left the boy’s hand but before it touches the ground, and assume that forces exerted by the air are negligible. For these conditions, what are the force(s) acting on the ball?

- (a) A downward force of gravity along with a steadily decreasing upward force.
- (b) A steadily decreasing upward force from the moment it leaves the boy’s hand until it reaches its highest point; on the way down there is a steadily increasing downward force of gravity as the object gets closer to the earth.
- (c) An almost constant downward force of gravity along with an upward force that steadily decreases until the ball reaches its highest point; on the way down there is only a constant downward force of gravity.
- (d) An almost constant downward force of gravity only.
- (e) None of the above. The ball falls back to ground because of its natural tendency to rest on the surface of the earth.

Note the absence of terms like “projectile motion,” which might cause a language model (or a student!) to mechanically regurgitate the fact that only the force of gravity is acting. Despite this challenge, ChatGPT’s response, which was representative of its performance on these ten “entirely correct” free responses, was as follows:

- (D) An almost constant downward force of gravity only.

After the steel ball leaves the boy’s hand, the only force acting on it is the force of gravity, which is directed downwards. This force is almost constant throughout the motion of the ball, with only slight variations due to changes in the distance between the ball and the center of the Earth. There is no upward force acting on the ball, as the ball is not being propelled upwards once it leaves the boy’s hand. As a result, the ball will continue to rise until the force of gravity slows it down and eventually reverses its direction, causing

it to fall back to the ground. Therefore, the correct statement is (D) An almost constant downward force of gravity only.

In addition to identifying the correct force, it makes specific note of the absence of any ongoing propulsive force (For comparison, in the Fall ’18 post-test student data, 38% of students chose one of the responses which includes an upward force during the motion). We note also that ChatGPT goes out of its way to comment on and correctly explain why the correct answer choice has the qualifier “almost-constant” on the force of gravity. This level of commentary is in fact deeper than we would typically expect of a first-semester physics student, at least unprompted. In fact, the inclusion of this detail in the question strikes us as the kind of thing an instructor does to make sure that their questions are airtight so that the strongest students have no room to complain or overthink.

Of course, not all of ChatGPT’s responses reach this level of accuracy. There were five free-responses which we judged to be more correct than not (in fact, typically almost entirely correct), but containing at least one patently untrue statement which simply could not have been made by an expert physicist [31]. These minor errors were often “surprising” in the sense that they did not necessarily represent the kinds of classic misapprehensions we have come to expect from new students on similar questions. And the errors were often “unforced” in the sense that they did not appear in the portion of the explanation that was strictly necessary to identify the correct answer. For example, consider item 22 of the FCI, which asks about a rocket in deep space that was moving horizontally from left to right at a constant speed until its engine produced a period of constant upward force. The question asks about the speed of the rocket while the engine is on (whether it is increasing, decreasing, constant, etc).

ChatGPT correctly identifies that the speed is increasing while the engine is on, but says the following:

Since the rocket is subject to no outside forces, its motion is determined solely by the forces it experiences from its engine. When the engine is turned on, the rocket is subject to a constant thrust that is perpendicular to its initial motion. This thrust produces an acceleration that is also perpendicular to the rocket’s initial motion. As a result, the rocket moves along a circular path, with its speed increasing as it moves from point “b” to point “c”. Therefore, the answer is (B) its speed is continuously increasing.

All of the elements of the correct chain of reasoning are present: (1) the net force in this case is equal to the force of the engine (2) when the engine is on, the net force is nonzero (3) a nonzero net force causes an acceleration and (4) the acceleration is not directed against

the motion and hence it will cause the velocity to increase. But ChatGPT make an unforced error by trying to describe the nature of the rocket’s motion during this time, even though this was not part of the question. It identifies the rocket’s motion as “circular,” perhaps because in its corpus of physics there is a strong association between discussions of circular motion and “forces which are perpendicular to the direction of motion.” Missing the obvious detail that this for is only *initially* perpendicular to the motion, it draws an incorrect conclusion. And the mistake is as surprising as it is unnecessary: we do not imagine that many freshman physics students are picturing a rocket which becomes locked into something like cyclotron motion when they read this problem, even though many of the found it difficult (57% chose incorrectly on the Fall ’18 post-test).

These five responses correspond with the other five problems for which ChatGPT chose the correct multiple-choice answer, although we graded these responses separately and it was not a foregone conclusion that it would coincide in this way. In some ways perhaps it is not surprising that ChatGPT’s “fully correct” and “nearly correct” free-responses also produced it’s entire set of correct multiple-choice responses, but in our opinion, this can also be reasonably interpreted as a way in which ChatGPT “projects understanding”—its chain of reasoning seems actually correlated with its final answer, as we would expect from a human being that was actually using the chain of reasoning to guide them through their thinking until they arrive at a final answer. This is striking because, to the best we can know, this is likely *not* how ChatGPT arrives at its multiple-choice responses.

After 10 fully-correct responses and five nearly-correct responses that nevertheless contain unambiguous errors, ChatGPT gives eight responses which are nowhere near the mark. These responses typically contain an assortment of words and phrases which are genuinely relevant to answering the problem. Often, entire sentences form accurate statements about the underlying physics. But the ideas are linked up in ways which are grammatically correct but often logically unreasonable or inconsistent, and the responses typically also contain discussions of details that are wholly irrelevant, intermixed on equal footing with the key concepts. The analyses may swerve unpredictably onto different tracks, or arrive at conclusions entirely at odds with the preceding arguments, as if the bot had no idea what it had just said, and yet it continues confidently to claim an inescapable conclusion. In short, these free responses are a masterful example of an ancient human art which shares its initials with the Bachelor of Science degree [32]. The impression given is similar to a student who is desperately brain-dumping all the knowledge they have in an attempt to gain some partial credit from their professor. They certainly do *not* project anything like understanding—quite to the contrary, if given by a human they would be taken as specific evidence that they are regurgitating memorized words and do not know what they are talking about.

In ChatGPT’s defense, half of these eight problems in which the free-response is manifestly indefensible are problems that originally contained figures, and have been reworded for this work. In particular, these four problems are the ones in which the answers rely most on establishing some sense of direction, and ChatGPT is simply unable to recognize the directional system being used. On multiple occasions, for example, it speaks of an object traveling “North” as feeling a force of gravity which causes it to curve to the West, and makes other similar mistakes. This pattern of particular confusion may suggest that we need to consider other methods for encoding these problems without figures, since the errors seem more to show that ChatGPT cannot demonstrate an understanding of *direction* rather than say anything about its understanding of physics. But note that, when cardinal directions are not involved, ChatGPT actually does quite well on problems which we rephrased to circumvent the need for an explicit figure.

A representative case of a “thoroughly confused” response (without raising the issue of cardinal directions) occurs with the very first item on the FCI, which asks about the time it takes for two balls—one heavy and one light—to reach the ground when dropped from a roof. ChatGPT responds incorrectly that it will take “considerably less time” for the heavy ball to hit the ground. It gives its reasoning as follows:

The time it takes for an object to fall to the ground depends only on its initial height and the acceleration due to gravity. The mass of the object does not affect the time it takes to fall. Both balls are dropped from the same height at the same time, so they will hit the ground at the same time. However, the heavier ball will fall faster because it experiences a greater gravitational force due to its larger mass. In fact, the heavier ball will accelerate at the same rate as the lighter ball, but it will have a greater terminal velocity when it reaches the ground due to its greater mass. Therefore, the time it takes for both balls to hit the ground will be considerably less for the heavier ball, but not necessarily half as long. The exact difference in time will depend on the initial height of the building and the difference in mass between the two balls.

All the relevant ideas are present, and there are many true statements. But they are combined in incoherent and contradictory ways. It begins with a claim that the time to fall depends only on the initial height and the acceleration due to gravity, which is correct within the approximation of negligible air resistance intended in the problem, and should be enough to answer the question correctly. And indeed, ChatGPT makes the statement that “both balls hit the ground at the same time.” But then it swerves to add something untrue (that the heavy ball will fall faster because it experiences a greater gravi-

tational force) and something true but inconsistent with its prior analysis: the heavy ball *will* reach a larger terminal velocity if air resistance is non-negligible and both balls have comparable drag coefficients. It would seem that ChatGPT is mashing up ideas in its training text from similar problems whose purpose is to explore air resistance and similar problems whose purpose is to discuss the mass-independence of the gravitational acceleration of an object. It has no awareness that these two concepts are in tension.

We may grant that, in fact, the problem is slightly ambiguous because it does not tell the reader with certainty whether air resistance is to be regarded as negligible. As an experiment, we offered ChatGPT another bite at the problem in a new conversational instance with no memory of this initial discussion, and added the phrase “assume no air resistance” to the problem stem. This time ChatGPT responded:

The correct answer is (B) about half as long for the lighter ball as for the heavier one. The key to solving this problem is to understand that the acceleration due to gravity is the same for both balls. This means that both balls will fall at the same rate, regardless of their weight. The time it takes for an object to fall is determined by the distance it falls and the acceleration due to gravity. Since both balls are dropped from the same height, they will fall the same distance. The only difference between them is their weight. According to the formula for the time it takes an object to fall ( $t = \sqrt{2d/g}$ ), the time it takes for an object to fall is inversely proportional to the square root of the acceleration due to gravity ( $g$ ). Since the acceleration due to gravity is the same for both balls, the time it takes for the lighter ball to fall will be about half as long as the time it takes for the heavier ball to fall. Therefore, the correct answer is (B) about half as long for the lighter ball as for the heavier one.

This starts out promisingly but veers wildly off-track when it comes time to draw a conclusion.

Curiously, when we revisited this problem later, and once again offered the problem in a new conversational instance (including the “assume no air resistance” clarification), the bot finally arrived at the correct answer and defended it with admirable clarity:

(C) About the same for both balls. Since there is no air resistance, both balls will experience the same acceleration due to gravity, which is approximately 9.8 meters per second squared. The time it takes for an object to fall a certain distance under the influence of gravity is determined solely by the distance and the acceleration due to gravity, and not

by the object’s mass. Therefore, both balls will take the same amount of time to reach the ground. This can also be seen by considering the equation for the time it takes an object to fall a certain distance, which is  $t = \sqrt{2d/g}$ , where  $t$  is the time,  $d$  is the distance, and  $g$  is the acceleration due to gravity. Since the distance is the same for both balls (i.e., the height of the building), and the acceleration due to gravity is the same for both balls, the time it takes for each ball to reach the ground will be the same.

If this had been the bot’s initial response during the full round of assessment, it would have been classified in the “exactly correct” category.

### 3. Stability

As the preceding example shows, ChatGPT is more than willing to change its mind under the right conditions. While much more work could be done on this topic, we make a brief study of the “stability” of the chatbot’s responses by experimenting with two types of perturbations.

First, we make use of ChatGPT’s “regenerate response” button, which allows us to simply request that the model start over its process with the exact same prompt, albeit with some variation in whatever internal parameters it uses to traverse its vast network of possible responses. This is analogous to asking a numerical equation solver to start over with a different random initial guess, and just as in that case, one would hope that a stable method would consistently produce comparable outputs.

For this first, initial probe of the Chatbot’s stability, we regenerated each response just three times, but already some patterns seemed clear. The only cases where the bot consistently changed its answers between regenerations were a subset of the eight questions for which we judged that its written responses were incoherent. There were two other cases of correct answers where the model occasionally changed its mind when the responses were regenerated, but in those cases it stuck with the correct answers a clear majority of the time. This pattern is perhaps somehow comforting or impressive for the question of ChatGPT’s ability to project understanding: when it knows the answer, it knows the answer with some stability. When it is flying by the seat of its pants, it also does not care what destination it flies to.

The other form of perturbation which should be explored is perturbations to the input, rather than to the initial starting conditions of the model themselves. Of course, because we cannot control the inner workings of the model and put it back in the exact same internal state every time, it is not possible to disentangle these two. Nevertheless, we attempted for a subset of the problems

to feed ChatGPT variations on the same questions but with irrelevant words and sentence structures switched around (e.g. “A boy throws a steel ball straight up” becomes “a rock is tossed straight upward by a girl). Once again, we found that this generally did not affect the response if the bot got the initial problem right, but it did throw things for a loop when ChatGPT was “winging it.” This stability study was hardly exhaustive or fully rigorous but seems to at least point to some underlying features of the model’s sensitivity to inputs and internal conditions.

### B. “RANKED” responses

As discussed above in Sec:III, we also explored use of an alternative prompt structure suggested in ref [9]. Those authors found a markedly improved performance when ChatGPT responded to multiple-choice questions on the bar exam if they asked it to rank its top three choices instead of simply choosing one. This approach was theorized to blend two of ChatGPT’s strength: rejecting some clearly wrong responses via “non-entailment” (perhaps because they contain sequences of words that it predicts as highly uncorrelated with the topic) and probabilistic entailment (ranking options based on plausibility, rather than commit to a single correct response).

We repeated our administration of the FCI using the “RANKED” prompt described above, inspired directly from ref [9]. Interestingly, while a nine of ChatGPT’s answers changed using this approach, four answers which were previously correct flipped to incorrect, three answers which were previously incorrect flipped to correct, and two incorrect answers flipped to other wrong answers. The result, then, was actually a small but negative change in ChatGPT’s performance, as it finished with 14/23 questions correct.

There is considerably room to explore the prompt engineering used here, including particularly in the ranked-choice case. Note for example that the multiple-choice questions used in Ref [9] contained four options, and ChatGPT was asked to rank three. Perhaps for the FCI, where each item contains five options, we should have asked ChatGPT to rank the top four. Perhaps also a thorough study of ChatGPT’s explanations provided from the “RANKED” prompt would reveal some pattern that could either be amplified or minimized as desired through variations in prompt. We will leave these questions for future work, and simply note for the time being that both of the “straightforward” prompting schemes we tried (that is, prompting schemes which did not require an elaborate prompt engineering study to optimize) produced comparable results. It is tempting, if slightly navel-beholding, to speculate why the “RANKED” prompting scheme “improved model performance substantially” on the bar exam [9] and yet had a small detrimental effect here. Of course, the two re-

sults might be entirely the product of the chaotic and unpredictable complexities of a large language model in a way that sheds no meaningful light. But it also seems possible that this points to some relatively inherent difference in the way multiple-choice questions manifest in physics vs in law. Perhaps law, for example, for all its appearances of being black-and-white, lends itself more to cases where one option can be judged as “best” but where others display arguable shades of plausibility. Different circuit courts, for example, can come to contradictory conclusions about how to read a particular statute, even when both courts are composed of self-described “textualists.” Different lawyers can come to different conclusions about which motions are strategically appropriate to file. As such, even though items from the bar exam are surely written to have a clear-cut “best answer,” it may be the best by virtue of being “clearly the least unreasonable.” Distractor answer choices also may be more likely to be wrong by virtue of being irrelevant, since it is a worthwhile skill for lawyers to know which statutes and regulations do or do not apply to given conduct. In a conceptual physics question, by contrast, the right answer is correct not because it is the least implausible but because it is the only one which is logically compatible with the canonical laws of physics. Similarly, the distractors are more likely to contain all the same, relevant concepts, but challenge the student to identify which choice shows an airtight and self-consistent application of them. It may be that this deprives ChatGPT of the ability to gain ground by ruling out irrelevant distractors and ranking the plausible options probabilistically. But, much like ChatGPT itself, we are theorizing now outside the scope of our formal training. Our only true conclusion here is that ChatGPT’s ability to project understanding of conceptual physics seems relatively insensitive to the prompting format.

### C. “NOVICE” Responses

Of course, the above conclusions do not mean that ChatGPT will respond the same to every prompt. While it does not seem to matter a great deal in this context how we ask it to format its answers, we are able to change its responses by essentially asking it to “roleplay.”

This paper is focused on the question of whether ChatGPTs behavior is consistent with “understanding” introductory physics, but our motivation for asking that question is to understand how it may affect physics classrooms and physics pedagogy more broadly. One way it might be of use to an instructor would be as a way to gain insight into the unfamiliar mind of a novice physics student. If ChatGPT were able to successfully answer questions in a manner that plausibly mimicked typical student mental models [33] it might be of great value in testing and preparing lesson plans, refining exam questions, etc.

The question of whether ChatGPT can do this is the worthy subject of its own project, and one we hope to

pursue in subsequent work. But we make an initial stab at it here to establish some baseline results and investigate whether such deeper work is likely to bear fruit. To do this, we gave the modified FCI questions to ChatGPT again, but this time using a prompt which we developed which asks it to answer as though it had not yet studied any introductory physics. This prompt was developed through trial and error using a small subset of FCI problems for which we had strong evidence from our own students' prior pretests about how they might answer. These were items 4 and 26 from the FCI. Item 4 tests Newton's 3rd law in the unintuitive context of an asymmetric collision between a truck and a car. Prior to studying physics [34], many students (nearly 75% in our fall '18 pretest) believe that the truck will exert a greater force on the car than vice versa. We found that initially, even when we prompted ChatGPT to answer as though it had not studied Newton's Laws, it continued to give the expert answer that the forces were equal. It was only when we modified the prompt to include a reminder both before and after the question that we saw it give the infamous novice answer described above. We saw similar behavior in question 26, which tests concepts about the balance of forces in the context of kinetic friction. The prompt we ultimately chose was:

Please answer the following question as though you were a novice high-school student who has not studied physics and does not yet know or understand Newton's laws:

{question here}

Just give us your best guess. We know you may not know the correct answer, but we'd like to know which answer makes the most sense to you without any formal physics training.

We do not claim that this prompt is optimized in terms of generating plausible novice responses.

The results from ChatGPT were mixed. To quantitatively evaluate the performance, we scored its answers on a "key" comprising all of the most common responses given by students in our Fall '18 pretest data, 10 of which were already correct answers. ChatGPT's answers to questions with the "NOVICE" prompt matched these most-common student answers on 11 occasions out of 23. In particular, seven of these 11 matches came in situations where the most common pretest answer was itself a correct answer. While it is not inherently a bad to match in these cases, because an educator using ChatGPT to test out possible student responses would want to know about cases where even an untrained student is likely to get the question right, it is discouraging to see comparatively few cases where the plurality student pretest opinion was wrong and ChatGPT was able to identify the distractor they would find most compelling.

Curiously, the problem was not that ChatGPT did "too well" on the problem set. In fact, ChatGPT's

"NOVICE" responses to the 23 modified FCI questions were right only nine times. There were only three such occasions where ChatGPT got a question right which the pretest plurality got wrong. But clearly, in the remaining cases, it did not agree with the plurality about which wrong answer seemed most intuitive.

This binary, agree-or-disagree framework is a little unfair to ChatGPT. Imagine a scenario where 50% of students choose wrong answer "A," 49% choose wrong answer "B," and only 1% choose the correct answer "C." If ChatGPT were to offer answer "B" when roleplaying as a student, it would hardly represent a glaring failure. To capture some of this nuance, we propose a simple measure where ChatGPT is given a point for each answer, weighed by the fraction of students from our Fall '18 pretest who chose that answer (so, half a point if 50% of students chose the same answer). Summing these scores and then normalizing by the equivalent score of the student key itself (which represents the maximum possible number of points ChatGPT could score) gives the bot a score of 66%. As this number indicates, ChatGPT is certainly succeeding on some level in modifying its performance to capture the typical thoughts and responses of novice students. But it clearly also leaves room for improvement in this regard; note that if we run this calculation for the average performance of someone guessing randomly, for this particular distribution of student responses the minimum score is still 56%.

Examining the free-responses of ChatGPT under the "Novice" prompt gives some insight into the reason it wasn't better able to mimic the behavior of an untrained student. A significant majority of its explanations referenced concepts like forces and accelerations that would likely not be the basis for a true novice's analysis of the problem. In fact, it routinely mentioned specific cases Newton's laws by name, despite the explicit instruction that it should answer as a student who "does not yet know or understand" them. It seemed instead to be answering from the perspective of a student who knew about Newton's laws, but who was having trouble applying them correctly.

We leave it for future work to see whether this feedback would allow us to further refine the prompt. However, we conjecture that it may be the nature of a LLM that it struggles to answer questions with a self-imposed blind spot. After all, ChatGPT presumably had instances in its training text where students mentioned Newton's laws, identified that they did not know how to use them, and proceeded to use them incorrectly. But it was less likely to have seen many cases where students mentioned Newton's laws by name for the purpose of saying that they had never heard of Newton's laws. As such, it might have struggled to identify cases where students were reasoning with a pre-Newtonian mindset based on an instruction like "you do not know or understand Newton's laws."

## V. CONCLUSIONS

The author has long believed that papers which title themselves with a question ought to be obligated to answer that question explicitly in the conclusions of their paper. Ours asks whether ChatGPT “can project an understanding of introductory physics,” [35], meaning, “can it display behaviors consistent with having an underlying understanding of kinematics and Newtonian dynamics, whether or not such underlying understanding actually exists?” The answer in brief appears to be “yes locally, not globally.”

In other words: in some isolated cases it responded to items from the FCI designed to test for conceptual mastery of introductory physics exactly the way an expert physicist might, despite (as far as we can tell) having never seen the question before and not having any specific programming dedicated to “doing physics.” The striking nature of this should not be overlooked, in an era where it can still be a struggle to get Siri or another voice assistant to correctly add an item to one’s calendar. And in our analysis, ChatGPT displays this ability about the same percentage of the time as a B- or B-level student taking a college physics course. This is considerably more proficiency than many physicists would have predicted was imminent just a few years ago.

On the other hand: when the mask slips, what it reveals is so clearly devoid of understanding that it spoils the charade. Like a student who started the required reading but did not finish it, ChatGPT’s best hope of projecting understanding is that you ask it something that it knows about and that the lunch bell rings before you can ask a follow up. It is not simply that at times it projects an understanding which is stronger at some times than at others: when it misses the mark, it misses by so much that it undermines the entire illusion. If a student submitted work which showed perfect mastery in one place and complete incoherence on the same set of topics immediately after, we would suspect cheating. ChatGPT may not be “cheating” per se, but it is not on the whole performing in a manner that one could confuse for full-fledged expertise.

We framed our purpose for this paper around four motivations, and the nuanced answer to our central question plays out differently for each. First, we asked what ChatGPT’s performance could tell us about the nature of some of the standardized assessment tools used in PER. One answer appears to be that it is important to treat these instruments as a whole, and not place too much emphasis on any single item. Assessments like the FCI were designed intentionally to probe central concepts along various different dimensions and with slight variations in the presentations of the concepts. ChatGPT’s oscillation between expertise and vacuousness underscores the necessity of this. A deeper study of the items that ChatGPT gets right, along with an investigation of its stability against perturbations in the question text, might also tell us more about which items or subtopics required

the deepest understanding to get right. And the results also remind us that there is yet another distinction to be drawn between “understanding” a concept, in the sense of “being able to apply it like an expert,” and *believing* a concept, in the sense of having an internal satisfaction that the concept is true because it should necessarily be true. This distinction is significant also among our students: compare how relatively easy it is for a student to learn that “both forces in the pair are always equal” in the context of Newton’s Third law with how few students actually feel that this fact is reasonable and intuitive when they first encounter it [36].

Second, we said that it was important to probe ChatGPT’s understanding because it (or tools like it) will soon be used by students in our classroom, whether we like it or not. The results here are mixed. On the one hand, ChatGPT’s inconsistent expertise means that students relying too heavily on its output in a physics class will be easy enough to spot, at least for the time being. But it is not hard to imagine how a student with modest expertise could supplement their own understanding by using ChatGPT as a sounding board for ideas. In fact, because its faulty responses stand out so much as being internally inconsistent, a student could conceivably ask it for help with every question and then learn the signs of when to disregard its answers. All this is to say, even at its current ability level, ChatGPT threatens the integrity of things like take-home tests, written homework assignments, and short-answer questions. It’s nontrivial and recently-augmented abilities with calculations and more mathematical questions only underscore this.

Third, we considered the possible role that ChatGPT could have as an in-class teaching tool. In some fields where its responses have a higher success rate, like computer science, some faculty are encouraging their students to come to class with a ChatGPT tab open, and to use it to ask brief clarifying questions, or find the bugs in their sample code, so that many more of these minor questions can be handled than a professor could hope to field on their own. At this juncture, we clearly do not recommend using ChatGPT this way in a physics classroom. Its hit rate for correct answers and correct explanations is simply too low, and the danger of confusing a student (or undermining their confidence) by giving them information that is only statistically trustworthy is too great. There may be some fundamental difference here between the fields: it strikes us that in coding classes, the fundamental unit of analysis and the ultimate goal of the work is to produce something that is essentially text: namely, a body of computer code. ChatGPT’s nature as an expert in creating functionally correct english text probably extends more naturally to producing “text” in a language like C++ than it does to parsing physics problems. Of course, as a counterpoint to its limitations, some ideas for classroom usage do suggest themselves, such as a “spot the bot” exercise where students compare answers from their peers and answers from ChatGPT, or perhaps an assignment or activity where a group works together to

document all of the errors in a faulty ChatGPT response. But such activities feel like novelties and are unsatisfying for their lack of long-term potential, since as artificial intelligence technology improves they seem likely to breakdown.

Finally, we asked about using ChatGPT as a tool to support physics instructors *outside* the classroom, by assisting with preparation. Here, we think that ChatGPT’s inconsistency may still be frustrating, but we can imagine several ways that it could still be a valuable if imperfect assistant. ChatGPT’s partial ability to produce responses while roleplaying as a novice (which improve when we allow more back-and-forth to remind the bot what it is and is not supposed to know) can help give insight as to which aspects of a topic need special attention in lecture materials. Its partial understanding and relatively literal mode of interpretation also make it an attractive tool to playtest possible exam questions. While the author would never admit to it himself, he has heard stories of other faculty whose exam questions occasionally contain glitches, loopholes, or overdetermined facts, which can be quite troublesome to remedy if they are discovered after the exam is underway [37]. Finally, ChatGPT might function as a tool to help with the drafting of new exam or homework questions in the first place. Based on what we have seen here, we imagine that its outputs would range from excellent to awful. But there are many times during the exhausting and challenging

process of developing new exam materials [38] where it would be a relief to be merely editing and improving another’s faulty work, rather than continually starting from scratch.

All of these motivating questions immediately suggest areas for follow-up projects. We intend to pursue some ourselves, and encourage others to do so as well. And finally, we must end by noting that, while history contains faulty predictions about the timeline of AI development in both directions [39], the current pace and nature of the field suggests it will continue to advance rapidly. Indeed, the self-reinforcing nature of the field as AI models learn to train themselves, coupled with the current exponential growth of computing power, suggests that the field as a whole could advance at an exponential or even superexponential rate [40]. Concepts which at the time of this writing are dismissed as being just beyond ChatGPT’s capabilities may be old news by the time the reader finds this—particularly given the pace of academic publishing. This is all the more reason why continued work in this area is urgently needed.

## ACKNOWLEDGEMENT

The author thanks Dr. Mark Kissler of the University of Colorado Hospital for, among other things, suggesting the verb “projects” to solve our titular dilemma.

- 
- [1] OpenAI. Chatgpt: Optimizing language models for dialogue, 2022.
  - [2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
  - [3] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.
  - [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
  - [5] Eva AM van Dis, Johan Bollen, Willem Zuidema, Robert van Rooij, and Claudi L Bockting. Chatgpt: five priorities for research. *Nature*, 614(7947):224–226, 2023.
  - [6] The PyCoach. Chatgpt: The end of programming (as we know it), Dec 2022.
  - [7] Christian Terwiesch. Would chat gpt3 get a wharton mba? a prediction based on its performance in the operations management course. *Mack Institute for Innovation Management at the Wharton School, University of Pennsylvania*. Retrieved from: <https://mackinstitute.wharton.upenn.edu/wpcontent/uploads/2023/01/Christian-Terwiesch-Chat-GTP-1.24.pdf> [Date accessed: February 6th, 2023], 2023.
  - [8] Jonathan H Choi, Kristin E Hickman, Amy Monahan, and Daniel Schwarcz. Chatgpt goes to law school. Available at SSRN, 2023.
  - [9] Michael Bommarito II and Daniel Martin Katz. Gpt takes the bar exam. *arXiv preprint arXiv:2212.14402*, 2022.
  - [10] At the time of this writing, the paper making this prediction was two months old. We are eagerly watching the calendar.
  - [11] Jürgen Rudolph, Samson Tan, and Shannon Tan. Chatgpt: Bullshit spewer or the end of traditional assessments in higher education? *Journal of Applied Learning and Teaching*, 6(1), 2023.
  - [12] Thomas Rid. Five days in class with chatgpt, Jan 2023.
  - [13] Faq.
  - [14] These were its actual responses when I asked it to complete the sentences.
  - [15] There may be an exception for raw mathematics—see Methods section.
  - [16] Grant P Wiggins and Jay McTighe. *Understanding by design*. Ascd, 2005.
  - [17] Namely, that it certainly does not.
  - [18] Marvin L Minsky. Why people think computers can’t. *AI magazine*, 3(4):3–3, 1982.
  - [19] Julian D Gifford and Noah D Finkelstein. Categorical framework for mathematical sense making in physics. *Physical Review Physics Education Research*, 16(2):020121, 2020.

- [20] David Hestenes, Malcolm Wells, and Gregg Swackhamer. Force concept inventory. *The physics teacher*, 30(3):141–158, 1992.
- [21] Carl Wieman and Katherine Perkins. Transforming physics education. *Physics today*, 58(11):36, 2005.
- [22] Andrew Elby. Helping physics students learn how to learn. *American Journal of Physics*, 69(S1):S54–S64, 2001.
- [23] Jonathan Tuminaro and Edward F Redish. Understanding students' poor performance on mathematical problem solving in physics. In *AIP Conference Proceedings*, volume 720, pages 113–116. American Institute of Physics, 2004.
- [24] Popular among the instructors, at least, if not their students.
- [25] Bethany R Wilcox and Steven J Pollock. Investigating students' behavior and performance in online conceptual assessment. *Physical Review Physics Education Research*, 15(2):020145, 2019.
- [26] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- [27] Alexa Steinbrueck. Can chatgpt do image recognition?, Jan 2023.
- [28] Jing Han, Lei Bao, Li Chen, Tianfang Cai, Yuan Pi, Shaona Zhou, Yan Tu, and Kathleen Koenig. Dividing the force concept inventory into two equivalent half-length tests. *Physical Review Special Topics-Physics Education Research*, 11(1):010112, 2015.
- [29] Josh Zumbrum. Chatgpt needs some help with math assignments, Feb 2023.
- [30] Alan Van Heuvelen. Learning to think like a physicist: A review of research-based instructional strategies. *American Journal of physics*, 59(10):891–897, 1991.
- [31] Perhaps we should specify a suitably caffeinated expert physicist.
- [32] No, we are not going to spell it out even in the footnote, but thank you for checking.
- [33] Donald A Norman. Some observations on mental models. *Mental models*, 7(112):7–14, 1983.
- [34] and, sadly, in a good portion of the time *after* studying physics.
- [35] Colin G West. AI and the FCI: Can chatGPT project an understanding of introductory physics? *arXiv preprint*, 2023.
- [36] CH Poon. Teaching newton's third law of motion in the presence of student preconception. *Physics Education*, 41(3):223, 2006.
- [37] Or so we are told.
- [38] James H Smith and Alfred G Costantine. Writing better physics exams. *The Physics Teacher*, 26(3):138–144, 1988.
- [39] Stuart Armstrong, Kaj Sotala, and Seán S Ó hÉigearthaigh. The errors, insights and lessons of famous AI predictions—and what they mean for the future. *Journal of Experimental & Theoretical Artificial Intelligence*, 26(3):317–342, 2014.
- [40] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, Oxford, 2014.

---

# ATCO2 corpus

## A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications

---

Juan Zuluaga-Gomez<sup>\*,1,2</sup> Karel Veselý<sup>3</sup> Igor Szöke<sup>3,4</sup> Petr Motlicek<sup>1,3</sup>  
 Martin Kocour<sup>3</sup> Mickael Rigault<sup>5</sup> Khalid Choukri<sup>5</sup> Amrutha Prasad<sup>1,3</sup>  
 Saeed Sarfjoo<sup>1</sup> Iuliia Nigmatulina<sup>1</sup> Claudia Cevenini<sup>6</sup>

Pavel Kolčárek<sup>7</sup> Allan Tart<sup>8</sup> Jan Černocký<sup>3</sup>

<sup>1</sup>Idiap Research Institute, Martigny, Switzerland

<sup>2</sup>Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland

<sup>3</sup>Brno University of Technology, Speech@FIT, IT4I CoE, Brno, Czech Republic

<sup>4</sup>ReplayWell, Brno, Czech Republic

<sup>5</sup>Evaluations and Language Resources Distribution Agency (ELDA), Paris, France

<sup>6</sup>Romagna Tech, Forlì, Italy

<sup>7</sup>Honeywell, Brno, Czech Republic

<sup>8</sup>OpenSky Network, Burgdorf, Switzerland

\*Corresponding author: [juan-pablo.zuluaga@idiap.ch](mailto:juan-pablo.zuluaga@idiap.ch)

### Abstract

**Abstract:** Personal assistants, automatic speech recognizers and dialogue understanding systems are becoming more critical in our interconnected digital world. A clear example is air traffic control (ATC) communications. ATC aims at guiding aircraft and controlling the airspace in a safe and optimal manner. These voice-based dialogues are carried between an air traffic controller (ATCO) and pilots via very-high frequency radio channels. In order to incorporate these novel technologies into ATC, large-scale annotated datasets are required to develop the data-driven AI systems. Two examples are automatic speech recognition (ASR) and natural language understanding (NLU). However, ATC is considered a low-resource domain. In this paper, we make several contributions aiming at overcoming these disadvantages. First, we introduce the *ATCO2 corpus*, a dataset that aims at fostering research on the challenging ATC field, which has lagged behind due to lack of annotated data. Second, we open-source a GitHub repository<sup>1</sup> that contains data preparation and training scripts useful to replicate some of our baselines related to ASR and NLU for ATC communications. The *ATCO2 corpus* covers 1) data collection and pre-processing, 2) pseudo-annotations of speech data, and 3) extraction of ATC-related named entities. The *ATCO2 corpus* is split into three subsets. 1) *ATCO2-test-set corpus* contains 4 hours of ATC speech with manual transcripts and a subset with gold annotations for named-entity recognition (callsign, command, value). 2) The *ATCO2-PL-set corpus* consists of 5281 hours of unlabeled ATC data enriched with automatic transcripts from an in-domain speech recognizer, contextual information (list of relevant n-gram sequences per utterance), speaker turn information, signal-

---

<sup>1</sup>Our code will be stored in the following public GitHub repository <https://github.com/idiap/atco2-corpus>.

to-noise ratio estimate and English language detection score per sample. These two are available for purchase through ELDA in <http://catalog.elra.info/en-us/repository/browse/ELRA-S0484/>. 3) The *ATCO2-test-set-1h corpus* is a one-hour subset from the original test set corpus, that we are offering for free in the following website: <https://www.atco2.org/data>. We expect the *ATCO2 corpus* will foster research on robust ASR and NLU not only in the field of ATC communications but also in the general research community.

**Keywords:** Robust Automatic Speech Recognition, Natural Language Processing, Air Traffic Control Communications, Spoken Language Understanding, Signal Processing

## 1 Introduction

The corpus introduced in this research is within the domain of civil air traffic control (ATC) communications and management. ATC aims at managing the airspace in a safe and optimal manner. The communication is either via spoken or data-link messages, while the time-critical messages are always spoken. These communications involve an air traffic controller (from now on, ATCO) issuing spoken flight instructions to aircraft pilots during all phases of the flight. The dialogue follows a well-defined grammar and set of rules that ensures safety, reliability, and efficiency [1, 2]. This can be seen as a multi-speaker and multi-turn conversation.

Commonly, an ATCO addresses several pilots in a short period of time, which in turns becomes the main cause of increased workload and limiting factor to increase the overall system capacity, i.e., there is large space for optimization by only reducing ATCO's workload. A significant bottleneck in the pipeline is the considerable latency arising from an ATCO issuing a command by voice and inserting it manually into the ATCO's workstation (for control and record). Recent advances in automatic speech recognition (ASR) and natural language processing (NLP) technologies have opened new ways where ATCO's workload can be reduced<sup>2</sup> by integrating different systems in a cascade fashion. The systems for extracting the actual meaning from the original audio signal are commonly known as spoken language understanding (SLU).

Our previous works have made sizeable progress on independent systems for ATC, such as, robust ASR [3], NLP [4], and diarization and segmentation [2]. However, until today, these systems are close to non-existent in real-life ATC operations. In part, this is due to the intrinsic complexity of the task, and mainly to the lack of annotated data available to train these data-driven systems [5].

This paper introduces the *ATCO2 Corpus* derived from a joint contribution from Clean Sky 2 Joint Undertaking (JU) and EU-H2020. ATCO2<sup>3</sup> project developed a platform to collect, organize, pre-process and automatically annotate ATC dialogues<sup>4</sup>. The main bottleneck towards ASR or natural language understanding (NLU) techniques for ATC are the lack of annotated data. Further, its collection and annotation requires trained personal, thus, the whole pipeline becomes excessively costly and impractical. This study presents how the entire data collection and annotation process can be efficiently accelerated by using already existing machine learning (ML) concepts.

The overall *ATCO2 Corpus* ecosystem is depicted in Figure 1. We release two corpora targeted to ATC for research in robust ASR, NLP and NLU, i.e., the i) *ATCO2-test-set corpus* and ii) *ATCO2 pseudo-labeled set corpus* (*ATCO2-PL-set corpus*). The former contains word-level and named entities<sup>5</sup> gold annotations. In total, we release 4 hours of speech with various useful metadata (see the blue circles from Figure 1). The latter, *ATCO2-PL-set corpus*, contains ~5281 hours of ATC audio recordings, where each utterance includes a detailed set of metadata. For instance: pseudo-transcripts obtained from an in-domain ATC ASR system (the pseudo transcripts contain also diarization and segmentation information), contextual information (list of word sequences for lattice-boosting of

---

<sup>2</sup>In fact, workload reduction is also translated in reduced flight time, which decreases overall operational costs and the environmental impact of aircraft.

<sup>3</sup>AuTomatic COllection and processing of voice data from Air-Traffic COmmunications, website: <https://www.atco2.org/>.

<sup>4</sup>We believe this pipeline can be easily adapted to other applications, where data scarcity is a latent problem, but access to unlabeled/non-annotated data is permissible e.g., patient–physician dialogues.

<sup>5</sup>Our Named Entity Recognition (NER) classes are: Callsign, Command, Value and Unnamed Phrase. The NER labels can be used to train/test an SLU system for slot filling.

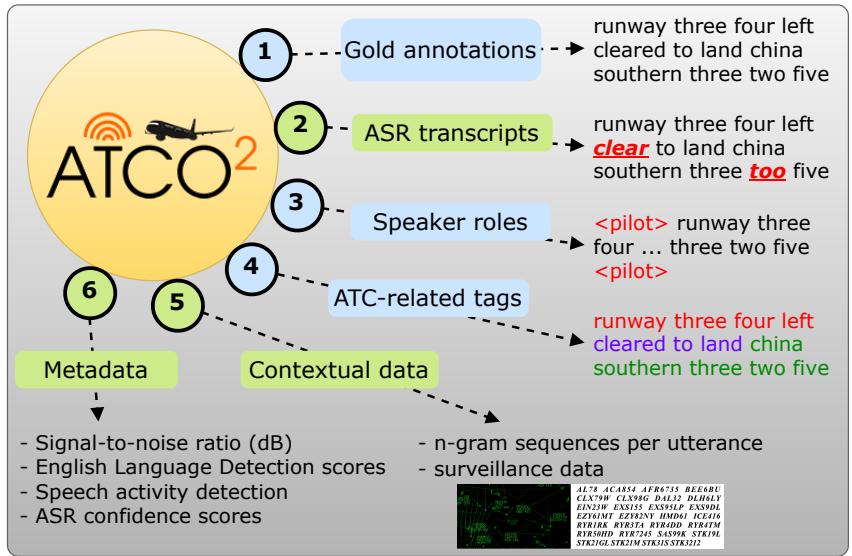


Figure 1: **ATCO2 corpus ecosystem.** Blue circles denote annotations only available for *ATCO2 test set corpus*. Green circles denote annotations and metadata available for both *ATCO2 test set* and *ATCO2 pseudo-labeled* corpus sets (see Table 1 bottom).

callsigns), signal-to-noise ratio (SNR) estimate, and English language detection (ELD) scores (see the green circles from Figure 1). Even though this is not the first publicly available corpus related to ATC communications [6, 7, 8, 9], to author’s knowledge, this is the first corpus that conveys annotated data for text and spoken-based tasks e.g., named entity recognition (NER), slot filling (SF), or sequence classification.

An overview of the data processing pipeline developed by ATCO2 and used to collect the *ATCO2 corpus* is depicted in Figure 2 (a more detailed description is in Section 4.3). The data processing pipeline consists of 1) speech pre-processing tools (segmentation, volume adjustment and discarding noisy recordings), 2) diarization (split audio per speaker), 3) ASR, 4) English language detection (ELD), 5) speaker role detection (SRD) e.g., ATCO or pilot, and 6) labeling of callsigns, commands and values with named entity recognition (NER). We used this pipeline to produce the *ATCO2-PL-set corpus* and *ATCO2-test-set corpus*, which cover more than ten airports worldwide. The ATCO2 corpus is publicly available in ELDA catalog at the following URL: <http://catalog.elra.info/en-us/repository/browse/ELRA-S0484/>. Further details about the data collection and pre-processing is covered in the Appendix B and our previous paper [2].

We also emphasize that the developed pipeline was not only used to collect the data presented in this paper, but also is running live at <https://www.spokendata.com/atco2>. The data is automatically fed, filtered, and pre-transcribed, so the amount of data will increase. We are searching for volunteers, both for feeding data from new airports (see Section 4.1) or for correcting the automatic transcripts (see Sections 4.2). In general, the *ATCO2 corpus* can be used to produce a robust ASR system for the ATC domain, and with its NER annotations it is possible to train models for SLU applications for extracting meaning from speech (e.g., callsign or command detection). However, we also believe that the pipeline developed by ATCO2 can be also adapted to data collection and annotation of different domains, e.g., call-centers conversations, or medical recordings.

**Motivation:** speech and text-based processing tools for ATC data could work better if we had a large amount of reliably annotated data. However, the collection and manual annotation requires qualified personnel, and it is costly. In addition, the recordings are often noisy (SNR below 15 dB), accented or with high speech rate (compared to conversational, read or spontaneous speech). Aligned to solve this, *ATCO2 corpus* answers four big challenges:

**Current ATC corpora are limited to automatic speech recognition.** However, ASR is only a small submodule of the whole pipeline and many more downstream tasks are indeed required, e.g., ATC-related NER or callsign detection and extraction. In our case, those are callsigns, commands

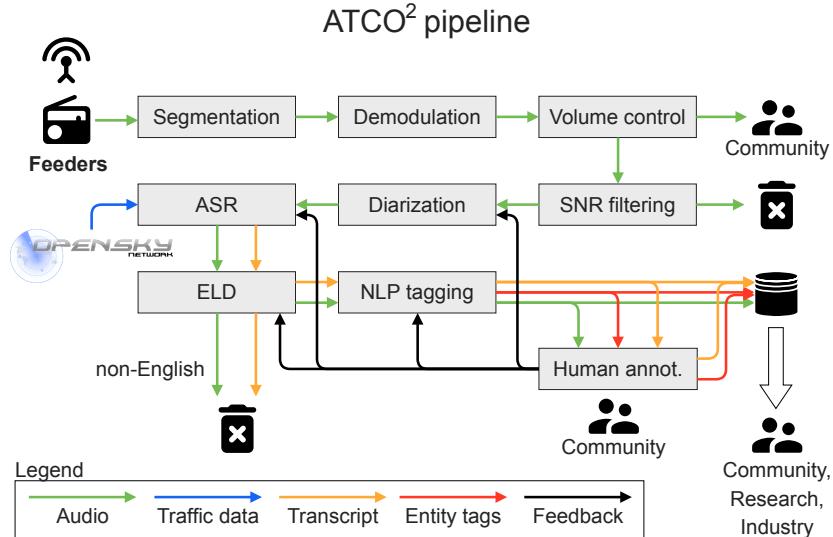


Figure 2: Data collection and data-processing pipeline developed in ATCO2 project.

and their values. *ATCO2 corpora* goes in this line and further by releasing gold annotations to train systems on ASR, NER, ELD, and SRD.

**Research on ATC communications has lagged behind due to the lack of annotated data.** The primary rational motive is the high annotation cost. ATC communications require eight to ten man-hours effort [5] to annotate one hour of raw controller-pilot dialogues. Primarily because it requires highly trained participants, often active or retired ATCOs. In total, after further pre-processing (e.g., silence and noisy segments removal) around one man-week work yields roughly an hour of annotations without silences [5, 10]. This number increases if further metadata is required, e.g., word-level tagging for NER. We address these issues by developing an efficient pipeline to collect, pre-process and automatically annotate ATC data (depicted in Figure 2). Using pre-transcribed data rather than transcribing from scratch reduced drastically the whole annotation process time period. Similarly, our observations during ATCO2 project revealed that the real-time factor (RTF) among the data transcribers varied drastically. For instance, untrained transcribers exhibited up to 50 RTF for transcribing ATC speech, including channel and NER tagging. However, trained transcribers reached as low as 20 RTF for the whole transcription process.

**Domain shift between ATC and non-ATC corpora is too strong.** Current ATC corpora contain data from only a few airports, and some were collected in clean and quiet simulation or training rooms [6, 7, 8]. Even though ATC speech should follow the same phraseology, the data from different airports substantially differ due to local conventions, speakers accent and rate of speech. All this together creates a considerable domain shift. Current ASR engines on the ATC domain are tailored to a particular airport<sup>6</sup>. Our ambition was, however, to collect and release annotated and pseudo-labeled recordings from many airports, which in turns can foster the training of more airport-agnostic ASR, NLU, and SLU systems. Moreover, data from non-ATC datasets like LibriSpeech<sup>7</sup> [13] do not match the ATC acoustics and its use does not help in the ASR training [3].

**Applicability on general spoken language understanding.** Even though the *ATCO2 corpus* and baselines presented here are aimed at a niche application (air traffic control communications), we believe that general-purpose research on NLU/SLU can widely benefit from this corpus. Most of the current benchmarks on SLU are widely saturated, where the performances (e.g., F1-scores) are near perfect, a couple of examples are ATIS [14] or SNIPS [15] datasets. Differently, *ATCO2 corpus* is composed of very noisy voice recordings (often below 15 dB SNR). The audio data is collected from devices (see Figure 4) owned by a community of volunteers (see Section 4.1), thus, it is more natural

<sup>6</sup>Other EU-funded projects, like Malorca or HAAWAI, only focus on developing ASR tools for one or at most two airports per project.

<sup>7</sup>This also includes other popular corpora, such as, CommonVoice [11] or Switchboard [12].

to find noisy data. This, in turn, increases the challenge of standard ASR systems, e.g., WERs of ~30% or above (see our previous baselines [16, 2]). We hope that the research community will build upon the *ATCO2 corpus* presented in this paper. Additionally, we hope that the presented baselines will foster research in the fields of ASR and NLP for ATC communications.

The paper is organized as follows. Section 2 covers previous work on ASR and NLP directed to ATC communications. Section 3 explains our proposed methodologies for the standardization of ATC communications annotation process. The data collection protocol, pre- and post-processing steps undertaken during the annotation process of *ATCO2 corpora* are described in Section 4. *ATCO2 corpora* data statics are reviewed in Section 5. Section 6 and 7 convey the proposed baselines on ASR and NLP, respectively. Section 8 covers the main legal and ethical implications of ATC data collection. Finally, we conclude this paper in Section 9 with final remarks and prospect of future work.

## 2 Previous Work on ATC Corpora

Currently, there is a huge diversity of databases related to speech and text tasks that have been promoting advances in artificial intelligence (AI). However, ATC communications are still considered an under resourced and underexplored area [16, 17]. Despite the growing interest in text and speech technologies for ATC, there is not a commercial ASR engine due to: (i) deficiency in terms of required performance (under 5% WER [18]), and (ii) lack of large-scale annotated speech data. The costly data collection and annotation makes it impractical, when transfer to a new airport requires data collection and annotation.

### 2.1 Background

Research seeking to aid ATCOs by ASR date as back as late 70s'. [19] proposed a system for isolated word recognition, speaker verification and commands recognition for military applications. Exploratory research towards integration of ASR technologies to aid ATCOs started in the late 80s with Hamel et al. [20]. Several other research directions cover user-friendly and robust automatic systems to train ATCOs, or the so called ‘pseudo-pilots’ [21]. Akin training systems have been proposed in [21, 22, 10]. We shortlist the three biggest European-based projects that aim at developing speech and text-based tools to aid ATCOs in their daily tasks. Initially, MALORCA<sup>8</sup> project was a step forward in demonstrating that ASR tools can cut down ATCOs workload [23] while increasing the overall efficiency [24]. Then, HAAWAI<sup>9</sup> project has led initiatives to extract key entities (e.g., NER or SF) in the transcribed dialogues produced by an ASR system [25]. Finally, ATCO2 project (our corpora) aimed at reducing the human work needed to develop ASR and SLU tools for ATC, mainly by integrating semi-supervised techniques in these systems [2, 4]. While the MALORCA and HAAWAI corpora are not public, in ATCO2 we developed a pipeline to collect large quantities of ATC speech data, which are distributed to the public through ELDA.

### 2.2 Command-related ATC Corpora vs Standard Corpora

The ATC speech corpora differ vastly from the standard ASR-training corpora. The root of the discrepancy is not only in grammar and vocabulary, but also in the audio quality. The standard corpora like Librispeech [13], Common Voice [11], AMI [26] or TED-LIUM [27] either target conversational, read or spontaneous speech while also being mostly regarded as ‘clean speech’. On the other hand, ATC corpora comprises considerably higher noise levels e.g., normally below 15dB signal-to-noise (SNR) ratio, heavily accented, high speech rate and artifacts. Previous work has demonstrated that the use of standard corpora do not bring significant improvement in ASR for ATC [4].

Even though, ATC English corpora share common grammar, there is still a domain shift caused by non-native speakers. One example are ATCOs from Switzerland. Even though they are from the same country, accent varies depending on the location. This, in turn, increases the challenge of developing

---

<sup>8</sup>MAchine Learning Of speech Recognition models for Controller Assistance, website: <http://www.malorca-project.de/wp/>.

<sup>9</sup>Highly Automated Air traffic controller Workstations with Artificial Intelligence Integration, website: <https://www.haawai.de/wp/>.

Table 1: Air traffic control communications related databases. This table list public and private ATC databases. The *ATCO2 corpora* are public databases.  $\dagger$ full database after silence removal.  $\ddagger$ speaker accents depend on the airport’s location, however, the accent of pilots are not known at any time of the communication due to privacy regulations.

Database	Details	Licensed	Accents	Hours $\dagger$	Ref
<i>Private databases</i>					
HAAWAI	Real data from Iceland and London airports	$\times$	Icelandic, British	47	[16]
MALORCA	Real data from Vienna and Prague airports	$\times$	German, Czech	13	[28, 29]
AIRBUS	Real data from Toulouse-Blagnac airport	$\times$	French	100	[30]
VOCALISE	Real data from terminal maneuvering area and area control center in France	$\times$	French	150	[31]
ENAC	Real data from two French en-route control centers and one major airport	$\times$	French	22	[32]
<i>Public databases</i>					
ATCOSIM	Simulated in studio, added cockpit noise. Recordings split by gender (Male/Female)	✓	Swiss German, German, French	10.7	[7]
UWB-ATCC	Real data from Prague airport	✓	Czech	13.2	[8]
LDC-ATCC	Real data from 3 US airports: Logan International, Washington National and Dallas Fort Worth airports	✓	American English	26.2	[9]
HIWIRE	Simulated in studio, ATC prompts, added cockpit noise	✓	French, Greek, Italian, Spanish	28.7	[33]
<i>Released corpora by ATCO2 project</i>					
<i>ATCO2 corpora</i>	Data from different airports and countries		Several $\ddagger$		
$\hookrightarrow$ ATCO2-test-set	Real data for ASR and NLP research.	✓	$\hookrightarrow$	4	
$\hookrightarrow$ ATCO2-PL-set	Pseudo-labeled real data for research in ASR and NLU.	✓	$\hookrightarrow$	5381	[2]
<i>Free access databases released by ATCO2 project</i>					
$\hookrightarrow$ ATCO2-test-set-1h	‘ASR dataset’: public 1 hour sample, subset of ATCO2-test-set-4h. <a href="https://www.atco2.org/data">https://www.atco2.org/data</a>	✓	$\hookrightarrow$	1	[2]
$\hookrightarrow$ ATCO2-ELD set	‘LID dataset’: public dataset for English language detection. <a href="https://www.atco2.org/data">https://www.atco2.org/data</a>	✓	$\hookrightarrow$	26.5	[34]

robust enough systems that generalize well across different in-domain environments. Therefore, a non-adapted ASR or NLP system will provide significantly worse performance due to unseen accents, out-of-vocabulary (OOV) words or simply due to discrepancy in the recording procedure.

Further details about the corpora produced by previous projects related to ATC communications are covered in Table 1. Current ATC corpora can be classified into two, public and private databases. Public databases normally require a small fee and sometimes are restricted to only-research purposes. While private corpora<sup>10</sup> are only usable along the concerned project, for instance, to train and test their ATC-related systems. One example is MALORCA, where the two produced corpora, *Prague* and *Vienna* datasets, are widely used by partners from HAAWAI and ATCO2 projects.

### 3 How To Transcribe Air Traffic Control Audio Data?

This section reviews our collective efforts to provide an unambiguous and clear protocol on how to annotate ATC speech data. We aim at avoiding as much as possible the errors caused by OOV words and phonetic dissimilarities (e.g., “hold in position” and “holding position”, or, “climb to two thousand” and “climb two two thousand”). We also rely on the International Civil Aviation Organization (ICAO), which defines a standard phraseology [1] to reduce these errors during the communications.<sup>11</sup> This section first formulates an approach to unify transcripts from different public

<sup>10</sup>In fact, nearly all ongoing and former projects in the area of ATC prohibit the release of databases, code, and AI models due to privacy issues.

<sup>11</sup>Previous work in [35] proposes a novel ontology agreed by several European institutions to annotate unambiguously these dialogues.

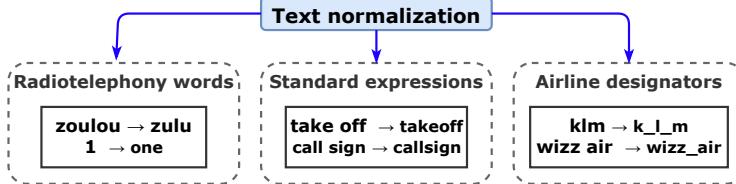


Figure 3: Text normalization applied to the transcription process of ATCO2 corpora. Further examples in Appendix C.

ATC databases (see Table 1 and Appendix C). Second, it discusses how to craft a lexicon tailored to ATC communications.

### 3.1 Unification of Transcripts

Differently from other corpora, ATC is regulated by a set of rules and a defined grammar. We have seen that in the already available databases (see Table 1), the transcription process and annotation rules widely diverge. There is not a clear path to follow when it comes to data collection and annotation. Even for a single database, it can be challenging to specify and follow their transcription conventions. One example are numbers. In ATC communications, numbers are key for addressing the aircraft or obtaining its speed or altitude. Several databases have opted to annotate numbers as digits (e.g., 1 → *I*), while other databases have chosen to use words (e.g., 1 → *one*). That is why the ATCO2 corpus also aims at providing a set of good practices and rules to correctly and unambiguously annotate ATC dialogues<sup>12</sup>. Therefore, if we succeed in reducing the variability of “writing the same thing in many ways”, we can considerably reduce the errors committed by subsequent systems in the pipeline, e.g., ASR or NLU. The next logic step, before starting the annotation process, is to define a set of rules to either, unify the transcripts of already available corpora<sup>13</sup> or, to annotate a new corpus.

In general, we apply three different text normalization approaches (see Figure 3) to foster good practices on the ATC-transcription process. We redirect the reader to Appendix C, where additional mapping rules are covered in Table 8. These mapping rules are applied as text filters. We use them to reformat the human-created gold transcripts for the ASR and NLP systems. For the annotation of *ATCO2-test-set corpus*, we also defined an annotation manual that is reachable from the transcription platform <https://www.spokendata.com/atco2>.

### 3.2 Lexicon

The lexicon is a table that maps words into pronunciations (phoneme-strings). It is a resource used by the HMM-based ASR tools. Our lexicon is based on the CMU Pronouncing Dictionary<sup>14</sup>, which defines the phoneme set, and which is used as the training data for the grapheme-to-phoneme (G2P) module that synthesizes pronunciations of “new words”. We gather all possible words from the training corpora, and we add some other words from different resources. We synthesize the pronunciations by G2P model trained with the Phonetisaurus<sup>15</sup> tool.

The ‘spelled acronyms’ like “KLM” (pronounced as “*k ey eh l eh m*”) are treated separately and represented as a single token (e.g., ‘k\_l\_m’). We also add manually created pronunciations for some non-English words that cannot be guessed by the G2P model. All the ‘word tokens’ in the lexicon are in lower-case. We keep only words relevant to ATC domain, i.e. words present in ATC transcripts or other resource. The lexicon contains 29k unique word-symbols.

Our strategy to mitigate the out-of-vocabulary problem is based on enriching the lexicon as much as possible as part of the data preparation. We enriched the lexicon with a list of airline designators

<sup>12</sup>Some of these rules and lessons can be easily adapted to other databases in the domain of ASR or NLU.

<sup>13</sup>We use these rules to normalize the transcripts of *UWB-ATCC* and *LDC-ATCC* databases (see Table 1 and 2) for experimentation.

<sup>14</sup>Dictionary at: <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>.

<sup>15</sup>More information in their GitHub repository: <https://github.com/AdolfVonKleist/Phonetisaurus>.

for callsigns (partly manually updated).<sup>16</sup> Also, we added all five-letter waypoint<sup>17</sup> names in Europe retrieved from open-source project Traffic<sup>18</sup>. Finally, we introduced some additional words, such as countries, cities, airport names, airplane models and brands, ATC acronyms, etc.

## 4 Data Collection

This section describes the collection and pre-processing of the audio data and ATC metadata in the ATCO2 corpus. An overview of the data processing pipeline is given in Figure 2. First, the data is collected via very-high frequency (VHF) radio receivers that are owned by a community of volunteers. Then, the audio and metadata are uploaded to OpenSky Network (OSN) servers<sup>19</sup> via Internet. Finally, the collected data is processed on a ReplayWell server<sup>20</sup> via REST API, and part of it is selected for human annotation (see Appendix A). The ReplayWell server hosts a major part of the data processing pipeline. We also rely on a community of volunteers for annotation.

### 4.1 Data Feeders

The data feeders are volunteers who capture ATC voice and upload it to OSN servers. The Data Feeders are typically aviation enthusiasts with possible prior operational experience, or people with an interest in aviation technologies (e.g., people doing domain related research, radio amateurs, etc.). To become a feeder, one needs to own a VHF receiver, which consists of an antenna, software defined radio (SDR) and a computer connected to the Internet. Affordable and popular options such as an RTL-SDR dongle and Raspberry Pi single board computer work sufficiently well in most cases. Quality of recorded ATC data varies depending on the equipment utilized during its collection (properly tuned gain parameter, position of antenna, DSP processor in the radio receiver). As an example, an affordable setup can be built with a *Sirio Md 118-137* antenna and an *RTL-SDR* radio receiver dongle (RTL2832U with 8-bit analog-to-digital converter), this setup is similar to items in Figure 4. For better quality, we recorded with a *Watson WBA-20* antenna and a *SDRPlay - RSP1A* radio receiver, which has a 14-bit analog-to-digital converter.

In some countries, it might be prohibited by law to record air traffic management (ATM) related data. The data feeders should check the applicable regulations before recording and feeding the data to the Internet. If you are interested in becoming a data feeder, please follow the instructions in the ‘feeder zone’ website: <https://ui.atc.opensky-network.org/set-up>

### 4.2 Data Annotators

The annotators are people who produce transcripts of ATC voice communications. These annotations also include assigning speakers roles and tagging of named entities. During the ATCO2 project, we relied on both the volunteers and paid transcribers. Volunteers with knowledge of ATC phraseology are ideal, but not strictly required.

Currently, we use our data processing pipeline (see Figure 2), which generates the initial transcripts and NLP tags. Pre-transcribing with AI tools speeds-up the overall transcription process. If you are interested in becoming an annotator, please create an account in the SpokenData transcription platform: <http://www.spokendata.com/atco2>. All the transcribed data within ATCO2 project’s life was packaged and released as the *ATCO2-test-set corpus*. Both the data feeders and annotators will have access to the data they provided.<sup>21</sup>

---

<sup>16</sup>List taken from Wikipedia: [https://en.wikipedia.org/wiki/List\\_of\\_airline\\_codes](https://en.wikipedia.org/wiki/List_of_airline_codes).

<sup>17</sup>A waypoint is an intermediate point or place on a route or line of travel, a stopping point or point at which an aircraft’s course is changed.

<sup>18</sup>Traffic project: <https://pypi.org/project/traffic/>.

<sup>19</sup>OpenSky Network is a non-profit association based in Switzerland. It aims at improving the security, reliability, and efficiency of the airspace usage by providing open access of real-world air traffic control data to the public. The OpenSky Network consists of a multitude of sensors connected to the Internet by volunteers, industrial supporters, and academic/governmental organizations.

<sup>20</sup>The whole pipeline runs live in the following URL: <https://www.spokendata.com/atco2>.

<sup>21</sup>More information on the official Opensky Network website: <https://opensky-network.org>.

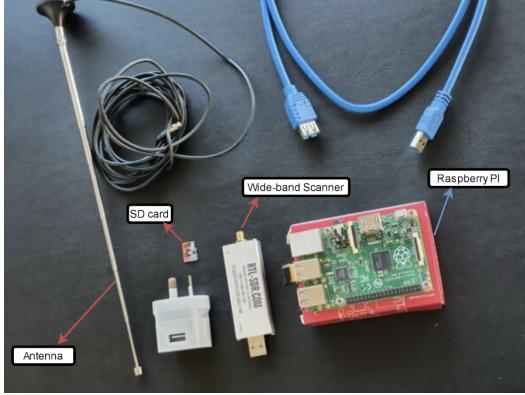


Figure 4: A set of items needed to set up a VHF receiver.

### 4.3 Data Processing Pipeline

The steps from our automatic *data processing pipeline* in Figure 2 are briefly described here:

**Segmentation and demodulation:** the RTL-SDR radio receiver tuned to a frequency provides a data stream in IQ format. The RTL-SDR software<sup>22</sup> has an in-built Voice Activity Detection (VAD) segmentation features. This is based on detecting abrupt changes of energy in the signal. The IQ signal is demodulated into a wave file with *csdr* software.<sup>23</sup> The *csdr* software is configured to remove DC offset, and we don't use automatic gain control. It is important to tune the *gain* parameter in the RTL-SDR software, so the audio is both well audible and not overboosted.

**Segment-based gain control:** the signal from distant airplanes can be weak. We noticed speaker turns are often separated by spikes in the waveform. The spikes arise from the window-based DC-offset removal in *csdr*. We detect these spikes and increase volume in segments separated by the spikes when needed.

**Signal-to-noise ratio filtering:** the next processing step is “signal-to-noise ratio filtering”. We discard recordings that are too noisy, but audio files with moderate level of noise are not discarded. We use WADA-SNR (Waveform Amplitude Distribution Analysis) [36] to estimate the SNR. WADA-SNR is based on analysis of shape of distribution over samples in a speech waveform. The non-speech parts are removed by a speech activity detection (SAD) tool [37] with a ‘tight’ preset, leaving almost no non-speech parts marked as speech.

**Acoustic-based speaker diarization:** a single recording can have multiple speakers in it, so the per-speaker segments are identified by diarization. We do it before the automatic transcripts are generated, so the NLU modules always process segments of a single speaker. Also, the annotators are asked to eventually rectify the per-speaker segments.

For details of the acoustic diarization VBx model, the reader is referred to [38]. This model uses a Bayesian hidden Markov model (BHMM) to find speaker clusters in a sequence of x-vectors. The x-vector extractor uses DNN architecture based on ResNet101 [38]. In the first step, Agglomerative Hierarchical Clustering (AHC) is applied to the extracted x-vectors. Then, Variational Bayes HMM over x-vectors is applied using the AHC output.

**Automatic speech recognition:** our ASR system has been trained on several publicly available databases [8, 9, 39, 7] and some private databases (AIRBUS, MALORCA). It is a hybrid ASR system trained with Kaldi. The system is covered in more details in Section 6, and also in [2]. The ASR output is confusion network, it is a ‘sausage-like’ structure with lists of alternate words in bins, and word-confidences in each bin sum up to one.

**English language detection:** we deployed an *English language detection* system (ELD) to separate non-English utterances from the input stream of data. Specifically, we used an NLP-based system that processes ASR output transcripts with word confidences. This system was more robust and better

<sup>22</sup>RTL-SDR radio receiver software: <https://github.com/szpajder/RTLSDR-Airband.git>

<sup>23</sup>CSDR software defined radio: <https://github.com/ha7ilm/csdr>

Table 2: Train and test sets configuration for baseline experiments.  $\dagger$ entire *ATCO2-PL corpus* used for training our ASR modules, see Table 1.  $\ddagger$ this subset filters out recordings that do not contain speaker role tags (used for speaker role detection). However, we report results on the full *ATCO2-test-set corpus* for the ASR experiments.

Dataset	Statistics			
	Nb. samples [k]	Duration [h]	SNR [dB]	Public
<i>ATCO2-PL-set (train)<math>\dagger</math></i>	3072	5281	any	✓
<i>ATCO2-test-set (test)<math>\ddagger</math></i>	3	3.4	$\leq 15$	✓

than standard acoustic-based ELD system [34]. Another benefit from using an NLP system is that it can jointly use logits or probabilities outputs from different ASR systems, which further can boost the results. Our ELD tool was previously covered in [2, 34].

**Post-processing by NLP:** in ATCO2 project, we focused on extracting knowledge from the text produced by the ASR system. *ATCO2-test-set corpus* contains rich metadata extracted with different NLP and NLU based modules. Specifically, we performed three tasks:

- *Callsign recognition*: locate the callsign and convert it to code such as "KLM91G"
- *ATCO/pilot classification*: decide who is speaking in the entire utterance
- *ATC-Entity recognition*: highlight the callsign, command and value entities in text

Further details about our NLP/NLU modules are covered in [2]. Information about integration and pre-processing pipeline is in Appendix B.

**Dataflow statistics:** the statistics for our *data processing pipeline* are accessible in <https://www.spokendata.com/atco2>. The daily numbers summarize the amount of recordings entering the pipeline, being rejected for various reasons (e.g., too low SNR, non-English language detected), being automatically processed, or selected for human annotation.

## 5 Datasets

Here, we describe in details the datasets for our baseline experiments. We evaluate on the *ATCO2-test-set corpus* as an in-domain test set, and *MALORCA-Vienna test set* as an unseen airport. The baseline systems are trained purely with the *ATCO2-PL-set corpus* and its automatic transcripts (i.e. pseudo-labels).

### 5.1 ATCO2 databases

**ATCO2-test-set corpus:** this dataset was built for development and evaluation of ASR and NLP technologies for English ATC communications. The dataset consists of English coming from LKTB, LKPR, LZIB, LSGS, LSZH, LSZB and YSSY airports. We provide two partitions of the data, the *ATCO2-test-set-1h corpus* and the *ATCO2-test-set corpus*. The first corpus contains 1.1 hours of open-sourced transcribed annotations, and it can be accessed for free in <https://www.atco2.org/data>. The latter adds around 3 hours of annotated data and the full corpus will be available for purchase through ELDA in <http://catalog.elra.info/en-us/repository/browse/ELRA-S0484>. The amounts of data per airport are summarized in Table 4. The recordings of both corpora are mono-channel sampled at 16kHz and 16-bit PCM. An example of the XML format for transcripts and tags is in Appendix D.

**ATCO2-PL-set corpus:** ATCO2 project recorded a large database of ATC voice communications. Altogether, we collected over 5281 hours of ATC speech from different airports around the world (see Table 3). In total, we cover ten airports. Table 3 depicts all this metadata per airport, further split by English language score. To the best of the author’s knowledge, this is the largest and richest<sup>24</sup> dataset in the area of ATC ever created that is accessible to the public. The automatic transcripts

<sup>24</sup>By richest, we mean quality of annotations and amount of metadata per sample. Also, this is the first public database in the area of ATC that targets NLU tasks.

Table 3: Stats about the collected databases per airport. Duration, SNR, language scores and contextual data columns report the mean and standard deviation (mean/std) per sample. Each recording/sample contains one or more segments (we provide timing information in RTTM format).  $\dagger$  abbreviation in IETF format.  $\ddagger$  total number of segments and accumulated duration of speech (after voice activity detection) per airport.

Database		Metadata				Contextual data	
ICAO - Airport	Accent $\dagger$	# Segments $\ddagger\dagger$	Dur. [sec]	SNR [dB]	Lang Score	# n-grams	# entities
<b>English Data (language score <math>\geq 0.5</math>)</b>							
EETN - Tallinn	et	79 k/131 hr	6.0/3.4	4.6/7.8	0.96/0.08	104/26	37/9
EPLB - Lublin	pl	<1 k/<1 hr	13.3/8.0	2.5/8.2	0.94/0.11	19/10	4/2
LKPR - Prague	cs	999 k/1762 hr	6.4/4.3	14.2/8.2	0.95/0.09	230/95	70/30
LKTB - Brno	cs	401 k/888 hr	8.0/14.4	4.1/15.7	0.88/0.15	49/35	15/10
LSGS - Sion	fr-ch	168 k/330 hr	7.1/4.8	10.0/8.0	0.87/0.15	56/23	20/8
LSZB - Bern	gsw	324 k/699 hr	7.8/5.0	15.4/10.7	0.90/0.13	101/42	36/15
LSZH - Zurich	gsw	470 k/921 hr	7.0/4.6	7.8/7.7	0.94/0.1	526/179	169/55
LZIB - Bratislava	sk	9 k/24 hr	8.8/6.9	5.4/8.7	0.86/0.15	68/27	22/8
YBBN - Brisbane	en-au	105 k/170 hr	5.8/4.1	10.2/5.8	0.93/0.1	268/86	95/30
YSSY - Sydney	en-au	49 k/77 hr	5.7/9.2	3.1/7.0	0.92/0.11	495/148	174/52
others - others	others	<1 k/<1 hr	5.0/6.7	4.0/7.4	0.92/0.11	55/260	16/78
<b>Non-English Data (language score &lt; 0.5)</b>							
EETN - Tallinn	et	2 k/2 hr	4.0/2.4	2.9/10.8	0.3/0.14	95/30	33/11
EPLB - Lublin	pl	<1 k/<1 hr	13.1/2.7	-8.4/12.8	0.2/0.13	17/7	4/1
LKPR - Prague	cs	105 k/187 hr	6.4/5.4	13.8/9.3	0.18/0.16	217/97	67/30
LKTB - Brno	cs	214 k/611 hr	10.3/19.2	6.5/11.8	0.15/0.15	56/33	18/10
LSGS - Sion	fr-ch	57 k/83 hr	5.3/3.6	9.8/9.3	0.27/0.14	56/25	20/8
LSZB - Bern	gsw	42 k/55 hr	4.7/3.4	13.6/13.9	0.30/0.13	102/45	37/16
LSZH - Zurich	gsw	36 k/49 hr	5.0/4.1	2.0/12.7	0.25/0.15	485/180	157/56
LZIB - Bratislava	sk	10 k/26 hr	9.0/7.6	7.1/7.0	0.18/0.15	72/26	23/8
YBBN - Brisbane	en-au	7 k/10 hr	4.9/4.8	5.9/12.3	0.24/0.16	268/79	95/28
YSSY - Sydney	en-au	3 k/3 hr	3.9/2.3	2.7/10.5	0.33/0.13	481/151	169/53
others - others	others	<1 k/<1 hr	5.2/6.4	3.7/9.0	0.26/0.15	0/0	0/0

Table 4: ATCO2-test-set corpora split by airports.

ICAO - Airport	ATCO2-test-set		ATCO2-test-set-1h	
	sentences	words	sentences	words
LKPR Prague	207	2686	102	1254
LKTB Brno	60	854	32	451
LSGS Sion	932	10183	256	2684
LSZB Bern	452	5908	172	2323
LSZH Zurich	640	8123	126	1764
LZIB Stefanik	165	2256	79	1051
YSSY Sydney	1065	10434	102	1058
Sum	3521	40444	689	10585

are stored as confusion network, stored as a *ctm* text format extended to have more words per line (confusion network bin):

```
<wav-id> <speaker> <t_begin> <dur> <word1> <conf1> <word2> <conf2> ...
LKPR_Tower_134_560MHz_20211223_154543 A 1.25 0.10 the 0.845 papa 0.042 ...
```

Another view of the data is in Figure 5, where we don't split by the English detection score. The distributions are from the full, 5281 hours dataset. In sub-figure 5a, we see that the majority of data was recorded in Prague, Brno, Zurich, Bern, and Sion airports. This is because we started recording these airport's data early in the ATCO2 project. Next, we also have some data from Brisbane, Talinn, Sydney, Bratislava (STEFANIK) and Lublin. In 5b, we see that majority of our data has high English scores (the dashed line 'ALL DATA'). There are some airports that have more non-English utterances: Brno, Bratislava (STEFANIK). And there are some airports, for which the distribution is more uniform than for others: Sion. We know for sure that local language is present at higher quantities in the data from Brno, Bratislava and Sion airports. And bigger airports usually have a policy to speak only in English, which explains low number of detections of non-English speech there. From 5c we

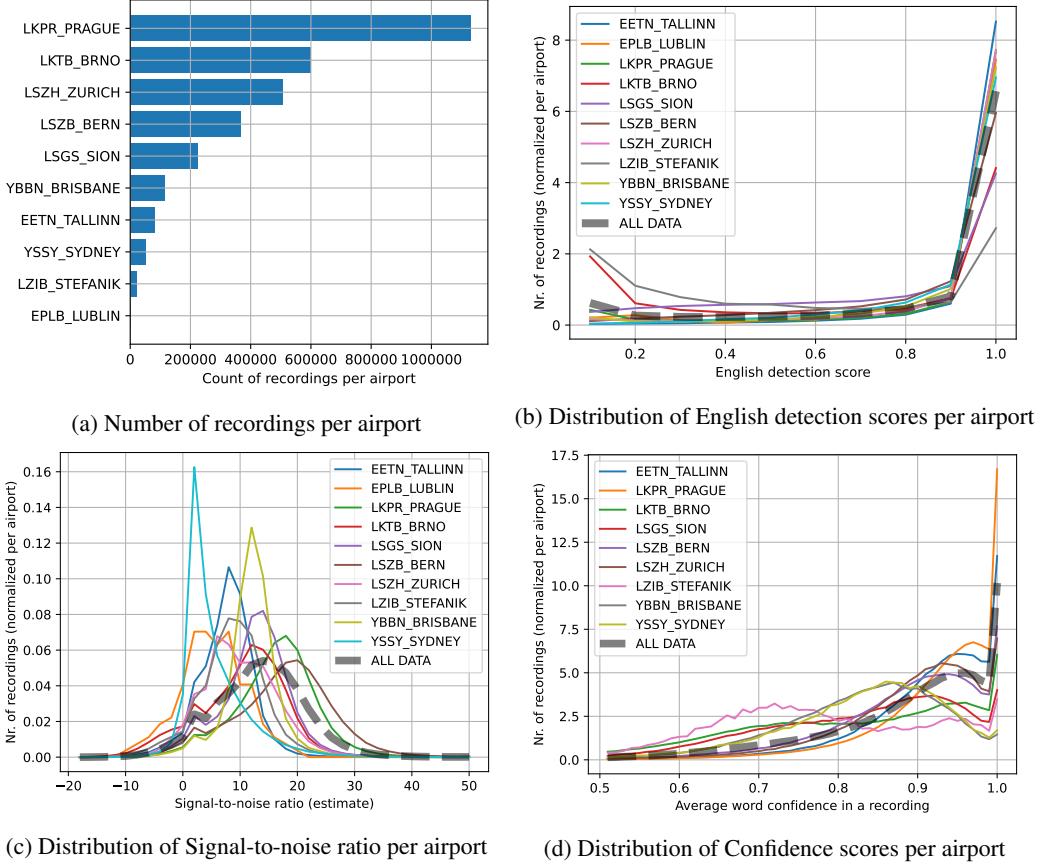


Figure 5: Distribution plots of metadata for *ATCO2-PL set corpus*.

see that levels of noise differ across the airports. The cleanest signal is for Prague and Bern, while high levels of noise are from Sydney and Lublin, and some noise also for Tallinn and Bratislava. This indicates that the recording setup could be improved. And in 5d are the distributions of confidences of the automatic transcripts, assigned by the seed ASR system. Majority of the probability mass is in interval (0.8, 1.0) (dashed curve ‘ALL DATA’). The highest confidence is assigned to Prague data (highest peak on right). The lowest confidence have the data from Bratislava, Brisbane and Sydney airports (distributions with leftmost modes). The higher tails with lower confidences are very likely caused by the non-English speech and noisy signal in the data.

## 5.2 Private Databases

**MALORCA Vienna test set:** The MALORCA Vienna test set is used in baseline ASR experiments as an unseen airport. No Vienna data are in the *ATCO2-PL-set* that use for training the acoustic model and language model. On the other hand, MALORCA Vienna data were present in the training of the seed system for generating the automatic transcripts. So it both unseen and indirectly seen at the same time. The set consists of ATCO speech only, which normally has lower error rates than the pilot speech [30, 40]. The total amount of speech after VAD is 1.9 hours. The audio data is mono-channel sampled at 8kHz and 16-bit PCM.

## 6 Automatic Speech Recognition

The Automatic Speech Recognition (ASR) system has an audio signal as its input and produces text transcripts as its output.

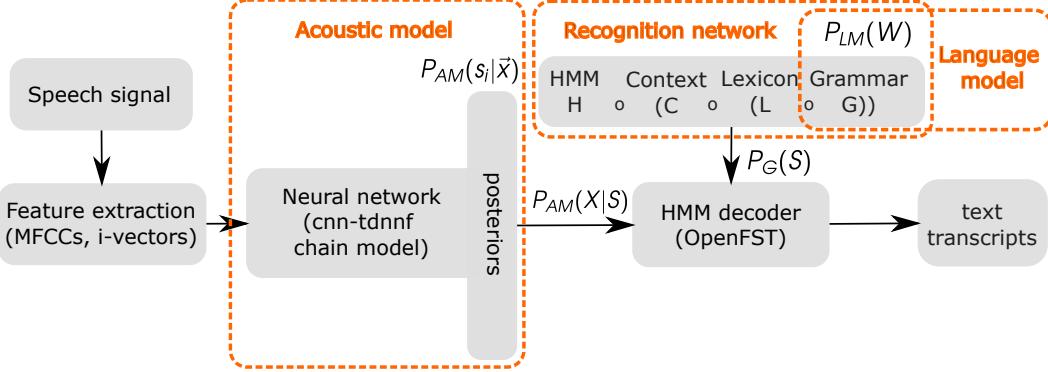


Figure 6: Hybrid-based ASR system. The inference pipeline consists of *feature extraction*, *acoustic matching* by acoustic model and *search* by HMM decoder that uses HCLG recognition network. On the output are text transcripts. Alternatively, the output can be a *lattice* (a graph with alternative decoded paths) or *confusion network* (time-sequence of bins with word-lists having scores).

At first, we trained a ‘seed ASR system’ from several existing ATC databases. The seed ASR system is part of the ‘data processing pipeline’ (see Section 4.3). This ASR produced the automatic transcripts for the *ATCO2-PL-set corpus* and also the initial transcripts for the *ATCO2-test-set corpus* that were manually corrected. During the ATCO2 project, we worked mainly with hybrid-based ASR systems, trained with the open-source toolkit Kaldi [41].

In hybrid speech recognizers, Hidden Markov Models support various speech rates, by allowing to “stay in a state” for some time via self-loop transitions. The acoustic scores come from a neural network, and the decoding process is based on searching in the matrix of acoustic scores for state-sequences with plausible transcriptions generated from a pre-compiled recognition network HCLG (i.e., a large HMM graph/model). Thus, HMMs provide a structure for mapping a temporal sequence of acoustic scores into a sequence of states [42, 43], from which the recognized words are extracted.

Inference in a hybrid speech recognizer has three stages: **feature extraction**, **acoustic matching** and **decoding**, the overall scheme is in Figure 6.

**Feature extraction** compresses the waveform into a sequence of fixed-length vectors of low dimension, in our recipe we use high-resolution MFCCs with i-vectors [44] appended.

**Matching of acoustic units** by acoustic model converts the input features into posterior probabilities of a closed set of acoustic units (phoneme states), whose time series forms the acoustic score matrix. In our recipe, we use ‘chain’ model neural network (NN) trained by Lattice-free MMI [45]. The NN topology is ‘cnn-tdnn’ architecture with 6 conv-relu-batchnorm-layer components followed by 9 tdnnf-layer components [46], and 2 softmax layers with 2000 outputs each. The acoustic model consists of 12.9 million trainable parameters.

**Decoding** searches for the most likely word sequence  $\hat{W}$  (transcription), in the matrix of acoustic scores. The search explores HMM paths that exist in a recognition network, termed HCLG graph. The standard decoding algorithm is based on two ideas: *token passing* and *beam search*. The search combines scores from the acoustic model, language model and lexicon, as shown in equation (1) :

$$\hat{W} = \text{wrds} \left( \underset{S}{\operatorname{argmax}} P_{AM}(S|\mathbf{X})^\kappa P_G(S) \right). \quad (1)$$

The acoustic model scores are the chain model posteriors  $P_{AM}(S|\mathbf{X})$ , where  $\mathbf{X}$  is the time-series of input features and  $S$  is an HMM state-sequence. The language model and lexicon scores are both represented in the graph score  $P_G(S)$  that is present in the HCLG recognition network.  $\kappa$  is an empirical scaling constant, for chain models the optimal  $\kappa = 1.0$ . And the function  $\text{wrds}(\cdot)$  is reading word-sequence from the state sequence  $S$  with the maximal score.

**The HCLG graph** is a Weighted Finite State Transducer (WFST). The HCLG graph is composed of a language model graph  $G$ , pronunciation lexicon graph  $L$ , context dependency graph  $C$  and phoneme

HMM graphs  $H$ . The HCLG graph contains graph costs  $P_G$  that originate from its source graphs, while the most important source is the language model. This was the description of a hybrid ASR system.

The other type of ASR systems are End2End systems. The End2End systems do not rely on HMMs and do not have a pronunciation lexicon. However, End2End systems require more training data to achieve good performance. Hybrid systems remain one of the best and more flexible approaches for building ASR engines. The HMM-DNNs based ASR are used in the current state-of-the-art systems for ASR in ATC communications [29, 28, 4, 3].

Hybrid-based ASR systems train independently the Acoustic model and the Language model. the language model is trained on a text corpus. This allows to incorporate text resources without the necessity to have the corresponding audio. The hybrid ASR relies on a word-based lexicon, and words that are not in the lexicon or language model cannot be hypothesized by ASR decoder (Out-of-vocabulary word problem – OOVs).

We use the same ASR system both for ATCOs and pilots. The training recipe and databases for our ‘seed ASR system’ (including the train sets in Table 1) are covered in [2, 3, 4, 47]. Briefly, we used AIRBUS, MALORCA Vienna, ATCOSIM, UWB-ATCC, LDC-ATCC, HIWIRE and N4-NATO databases. In total, these form a database of  $\approx 135$  hours. We augmented this database with noises captured from LiveATC. And the data were further augmented by speed perturbation. Due to data license issues with some databases, this ASR system can be only used for research.

In a later stage of the ATCO2 project, we experimented with **contextual adaptation** and **semi-supervised training**. We later integrated these technologies into the ‘seed ASR system’.

The **contextual adaptation** improves the accuracy of the ASR system by feeding-in a rapidly changing contextual information. Based on surveillance data, we suggested a list of nearby callsigns into the recognizer [48]. This was done by applying a boosting WFST graph to HCLG or lattice. In HCLG boosting, we give score discounts to individual words, while in Lattice boosting, the score discounts are given to word sequences. The lattice boosting was used also when generating the automatic transcripts for *ATCO2-PL-set corpus*. Also, in [49, 50] lattice boosting and language model boosting are explored.

The **semi-supervised training** was used to improve ASR accuracy by retraining the acoustic model [2] on a mixture of manually and automatically transcribed data. ATCO2 data with automatic transcripts were mixed with transcribed data from other databases. We used per-frame gradient weighting by word confidences to de-weight data with unreliable transcripts. We further performed experiments in [47]. Here, we applied callsign boosting when generating automatic transcripts for semi-supervised learning, and we obtained 17.5% relative WER improvement measured on the callsign words.

## 6.1 Baseline experiments

During the ATCO2 project, we collected 5281 hours of ATC audio data from several airports. We processed the data with our automatic pipeline (see Figure 2) that filters the data and produces automatic transcripts. Inside the pipeline, there is an ASR system that is described in Section 6 and also in our previous work [2].

The purpose of these baseline experiments is to demonstrate what can be achieved with the data we collected and released in ELDA catalogue. From these automatic transcripts, we can bootstrap and build a new ASR system, without having licensing problems that exist for some other databases (Table 1). We built a new language model from all the generated transcripts. And, we experimented with training acoustic models on various subsets of the audio data. We re-used the lexicon from the ‘seed ASR system’.

The baseline experiments are described in Table 5, where we computed Word Error Rate (WER) on three test sets: *ATCO2-test-set*, *ATCO2-test-set-1h* and *MALORCA Vienna* test set. Each model is tagged with the number from the first column of Table 5. The MALORCA Vienna test set represents clean ATCO speech from an unseen airport.

**Analysis of ASR systems from Table 5:** In 1) we built the acoustic model and language model on all the data in the ELDA package, including the data that the English detector identified as non-

Table 5: Performance on *ATCO2-test-set corpus*. The ASR system is built from the automatic transcripts of the *ATCO2-PL-set corpus*. We use two *ATCO2-test-set corpus* splits, and MALORCA Vienna [28, 29] as an unseen airport. The data filtering is done according to: ELD (English language detection), SNR (signal-to-noise) ratio, and CNET (average word confidence in the recording).

System	Training hours	WER			Data Selection Method			
		ATCO2 test-set (4h)	ATCO2 test-set 1h	MALORCA Vienna	ELD	SNR	CNET	Note:
1)	5281	22.3	15.8	11.1	any	any	any	all ATCO2 data
2)	4500	22.5	15.7	10.0	>0.5	any	any	remove non-English
3)	3600	22.5	15.8	9.3	>0.7	>0	>0.8	remove low-confidence
4)	1500	23.4	16.7	11.9	>0.5	>16	any	remove low SNR (<16)
5)	3600	22.4	15.4	9.0	>0.5	any	any	random from 4500 hour set
6)	2500	22.6	15.8	9.0	>0.5	any	any	random from 4500 hour set
7)	1500	22.5	15.8	10.6	>0.5	any	any	random from 4500 hour set
8)	500	22.5	15.7	11.0	>0.5	any	any	random from 4500 hour set
9)	135+700	26.6	18.6	4.8	-	-	-	seed system

English. In 2) we excluded the non-English data, and from now on, the Language Model is always trained from transcripts of this 4500 hours dataset (except for seed system). In 3) we set the filtering thresholds higher to >0.7 ELD (English detection), >0 dB SNR and >0.8 CNET score (average word confidence in a confusion network of recording). In WER results for 1) 2) 3), we see that the *ATCO2 test sets* results stay similar, while the WER for MALORCA test set improves with stronger data filtering. In 4) we realized that it is not a good idea to discard too much noisy data by filtering >16 SNR. Next, in 5) we randomly selected 3600 hours from the 4500 hours dataset. This was to cross-check with the filtering we previously used in 3). To our surprise, the results are marginally better when randomly selecting the data. Next, in 6) 7) 8) we continued randomly selecting subsets from the 4500 hours dataset. This degraded the performance of MALORCA Vienna on 7) 8) by up to 2% WER. From the results, we notice that WER for ATCO2 test-sets is nearly constant, except 4). It seems that WER in the automatic transcripts is an important factor. The automatic transcripts are used as training targets, and the WER in transcripts pre-determines the performance of the trained system. The amount of training data is possibly less important, however the generalization to a new airport (MALORCA Vienna) is better with larger volumes of training data of 2500 or 3600 hours in ASR systems 5) 6). For completeness, we also add WER of the seed system 9). The seed system had few percent higher WER for ATCO2 test-sets. For MALORCA Vienna, the seed system works as good as 4.8% WER, as MALORCA Vienna corpus was present in its training data.

## 7 Natural Language Processing and Understanding

Until the previous decade, research on ATC was directed at only transcribing as accurate as possible the dialogues between ATCOs and pilots. However, transcription is only one part of the story and further information, such as, entity highlighting (also known as intent and slot filling) or speaker role detection is imperative in real-life ATC control rooms. The process of parsing these high-level entities from ATC audio can be seen as SLU, or from text as NLU.

Previous work has already explained several NLP tasks in the area of ATC. For instance, [51] describes a set of entities and elements that are present in ATC communications that are of special interest, e.g., commands and instructions. The authors advise that a real-life system should be composed of an ASR module to obtain the word-level transcripts of the communication. Later, a subsequent system should extract ATC-related key entities and then parse them into a specific grammar. We redirect the reader to [35], which developed an ATC-structured grammar accepted by several European institutes. Furthermore, in [51], the process of extracting key entities from audio is summarized in an entire pipeline composed of three submodules. Namely, speaker role detection, intent classification and, slot filling (analogous to NER but on audio level). They aim at inferring the near-future air traffic dynamics, which can aid ATCOs in their daily task. In addition, this system can notice communication errors caused by one of the speakers, also known as hear or read back errors. Some exploratory work addressing NLP and NLU on the framework of HAAWAI and ATCO2 projects (see Table 1) is described in [52, 17].

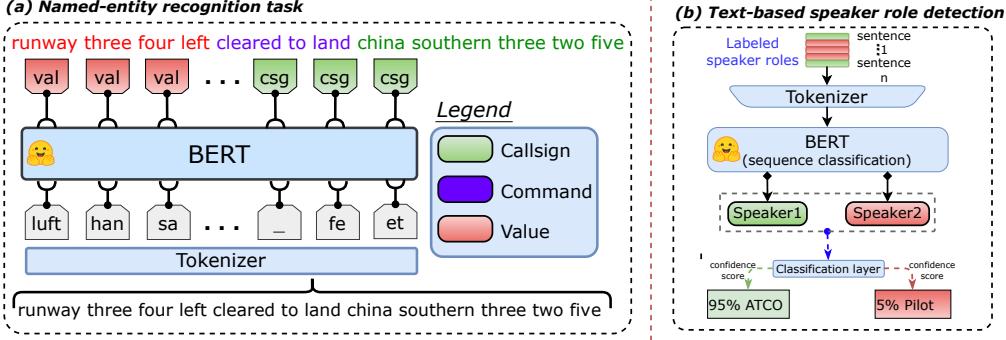


Figure 7: (a) Named entity recognition and (b) speaker role detection based on sequence classification (SC) for ATC utterances. Both systems are based on fine-tuning to ATC tasks from a pre-trained BERT [54] model. The NER systems recognizes callsign, command and values, while the SC assigns a speaker role to the input sequence.

In this section, we describe our baselines for two tasks related to NLP and NLU.<sup>25</sup> In ATC, in addition to transcripts generated by an ASR system, we can also extract rich metadata from the transcripts and audio. Some examples are (but not limited):

- ✓ What are the high-level entities in the communication? → named-entity recognition (NER) or slot filling (SF). Previous work in [49] and covered in Section 7.1,
- ✓ Who is talking? ATCO or pilot → speaker role detection (SRD), sequence classification. Early work on [52], and covered in Section 7.3,
- ✗ Is the pilot responding the correct information? → read-back error detection. Our previous work in [53],
- ✗ Is the communication being uttered in English language? → English language detection (ELD). Our previous work in [34].

We present baselines only on the above items marked with ✓, while the items marked with ✗ are, either, covered in previous work or are left as future research directions. Generally speaking, extracting the above-mentioned information could allow to further fulfill other ATC tasks, e.g., pre-filling radar labels in the ATC control room. Or, for example, decrease the workload of ATCOs and increase their efficiency by automating manual and effortful processes. In addition, reducing the overall probability of incidents and accidents due to air traffic management erroneous procedures is a supplementary by-product of introducing AI tools in the ATC control rooms.

## 7.1 Named Entity Recognition

Named entity recognition, or NER, is one of the most explored tasks in the field of information extraction and NLP [55]. NER aims to locate and classify entities in unstructured text into pre-defined classes or categories. Examples are, persons or organization names, expressions, or, for instance, callsigns or commands in ATC. Initially, NER was based on handcrafted lexicons, ontology, dictionaries, and rules [56]. Even though these systems were interpretable and understandable, they were prone to human errors. Collobert et al. [57] introduced machine learning-based methods for text processing in topics such as part-of-speech tagging, chunking, NER, and semantic role labeling. Further interesting works on NER are [58] focusing on multilingual NER for slavic languages, and [59] presenting a broad survey of NER methods. In practice, a NER system can be crafted by fine-tuning a pre-trained LM, e.g., BERT [54], RoBERTa [60], or DeBERTa [61]. Nonetheless, these models are data hungry and need expensive GPUs during its training and inference. Further work has been directed at reducing their computational footprint, by performing, for example, knowledge distillation [62].

<sup>25</sup>As we work on top of ASR transcripts, these tasks can be also cataloged as spoken language understanding.

Air traffic control communications typically carry structured information, including callsigns, commands and values. These can be seen as ‘named entities’. *ATCO2-test-set corpus* provides annotation on the word level that assigns pieces of text to these specific classes. We developed a baseline system to extract such information from ASR utterances, as depicted in Figure 7. An early implementation of this system was covered in [49]. However, these experiments were carried over private databases, so it is difficult to compare with our current results. That is why we base our experiments in [49], but we go beyond by open sourcing scripts to fine-tune a NER model with *ATCO2-test-set corpus*.

### 7.1.1 Experimental Setup

Our experiments are carried out with *ATCO2-test-set corpus* only, for both, training and evaluation.<sup>26</sup> The main reason is that none of the public databases from Table 1 contain NER annotations. As a workaround, we implemented a simple k-fold cross-validation scheme. We define  $K = 5$  folds, with a 70/10/20 ratio for train/dev/test subsets, respectively. We use ground truth ASR transcripts for training and testing NER.

First, we download a powerful pre-trained LM, BERT<sup>27</sup> [54], from HuggingFace [63, 64]. We append a linear layer with a dimension of 8 (following the IOB format, i.e., two outputs for each NER class) on top of the last layer of the BERT model. The model is later fine-tuned on the NER task, with each Fold  $K$  of the train splits. Each model is fine-tuned on an NVIDIA GeForce RTX 3090 for 10k steps. During experimentation, we use the same learning rate of  $\gamma = 5e-5$  with a linear learning rate scheduler. Dropout [65] is set to  $dp = 0.1$  for the attention and hidden layers, while Gaussian Error Linear Units (GELU) is used as activation function [66]. We also employ gradient norm clipping [67]. We fine-tune each model with an effective batch size of 32 over 50 epochs with AdamW [68] optimizer ( $\beta_1=0.9$ ,  $\beta_2=0.999$ ,  $\epsilon=1e-8$ ).

### 7.1.2 Results

We report the results obtained from the 5-fold cross validation experiments. We split the results by tags, namely, callsign, command and values. For each of them, we report precision, recall and F1-scores in Table 6. We obtained an average of 0.97, 0.82 and 0.87 F1-score for callsign, commands and values. We observed that the command class was the most challenging among the three classes. We believe this is because commands contain extra complexity in comparison to callsigns and values. For example, in some cases the ATCOs or pilots use several commands, or these are sometimes mixed in the same utterance. In contrary, callsigns follow a standard form, composed of an airline designator, numbers, and letters (spelled in ICAO phraseology). Values are composed of cardinal numbers and some standard words, e.g., ‘flight level’. We also noted a significant irregularity in performance for the command class between the 5 folds (see column: Command in Table 6). For example, worse → best scenario on F1-score was 0.79 → 0.85, almost a six-point drop. A five-point drop is also seen in precision and recall. These results are seen when comparing fold 2 (best) against fold 4 (worst).

In conclusion, the results from Table 6 are the first official baseline for NER<sup>28</sup> on the *ATCO2-test-set corpus*. However, there is room for improvement. For instance, implementing semi-supervised learning or data augmentation should bring robustness and yield higher performance. Similarly, one can pretrain the LM directly on ATC text rather than standard English text, which should bring in additional benefits. We leave this line of research for future work.

## 7.2 Callsign Recognition and Understanding

The Named Entity Recognition from previous section is capable to select words which form a callsign (i.e., highlight ‘swiss two six eight nine’). However, *ICAO Callsign Extraction* produces callsign directly in ICAO format (e.g., SWR2689), which is more useful for applications. This is not trivial because callsigns get commonly shortened, if the situation is obvious (e.g., ‘swiss two six eight nine’ → ‘six eight nine’, or ‘swiss eight nine’). And the underlying ASR produces errors in its automatic transcripts.

---

<sup>26</sup>We provide in the GitHub repository the utterance IDs splits utilized for these experiments.

<sup>27</sup>We use the pre-trained version of *bert-base-uncased* with 110 million parameters for all the experiments.

<sup>28</sup>After extensive research, to authors’ knowledge, this is the first official baseline on NER for air traffic control communications. We have not found any other work that is, both open-source and that targets NER.

Table 6: Different performance metrics for callsign, command and values classes of the NER system. Metrics reported for each of the 5-fold cross-validation scheme on *ATCO2-test-set corpus* with a `bert-base-uncased` model. @P, @R and @F1, refers to precision, recall and F1-score, respectively. Numbers in **bold** refers to the top performance per column among folds.  $\dagger$ mean score over the 5 folds.

Fold	Callsign			Command			Values		
	@P	@R	@F1	@P	@R	@F1	@P	@R	@F1
1	0.97	0.98	0.97	0.80	0.81	0.81	0.86	0.86	0.86
2	0.97	0.98	0.97	<b>0.83</b>	<b>0.86</b>	<b>0.85</b>	0.86	0.89	0.87
3	0.97	0.97	0.97	0.81	0.85	0.83	<b>0.87</b>	0.87	0.87
4	<b>0.98</b>	<b>0.98</b>	0.98	0.78	0.80	0.79	0.85	<b>0.90</b>	0.87
5	0.97	0.98	<b>0.98</b>	0.80	0.83	0.81	<b>0.87</b>	0.89	<b>0.88</b>
AVG $\dagger$	0.97	0.98	0.97	0.80	0.83	0.82	0.86	0.88	0.87

In the project, we explored two approaches. In [69], the ICAO callsign is retrieved by a BERT-based Encoder-Decoder neural network. This system directly takes outputs from an in-domain ASR system and extracts the ICAO callsign without relying on Named Entity Recognition as an intermediate step. The model uses a list of callsigns from surveillance data as context information, and it can return an ICAO callsign that is not present in the list. The overall approach is depicted in Figure 8.

The second approach [49] performs NER to extract the callsign within the sentence, which is later ranked by Levenshtein distance with the ones in the callsign list from the surveillance data. This approach always selects a callsign from the list. We showed that boosting callsigns with the combination of ASR and NLP methods eventually leads up to 53.7% of an absolute, or 60.4% of a relative, improvement in callsign recognition.

### 7.3 Speaker Role Detection

In NLP, text classification or sequence classification (SC) is a task that assigns a label or a class to a sequence of words [70, 71]. The hypothesis is that the words within the given text share a common role and meaning inside the sentence’s grammatical structure. One of the most acknowledged forms of SC is sentiment analysis, which assigns a label like positive, negative, or neutral to a sequence of text embeddings<sup>29</sup> [73]. Nowadays, state-of-the-art SC systems are based on the well-known Transformer, e.g., BERT [54] or RoBERTa [60]. Akin to NER, SC is considered a downstream task operating on ASR output.

In ATC, the dialogues are built on top of a well-defined lexicon and dictionary, which follows a simple grammar. This standard phraseology has been defined by the ICAO [1] to guarantee the safety and reduce miscommunications between the ATCOs and pilots. In this work, we propose some baselines on the SC task aimed at detecting the speaker role from transcribed ATC communications (sentences). Our previous work on speaker role detection is covered in [17, 52].

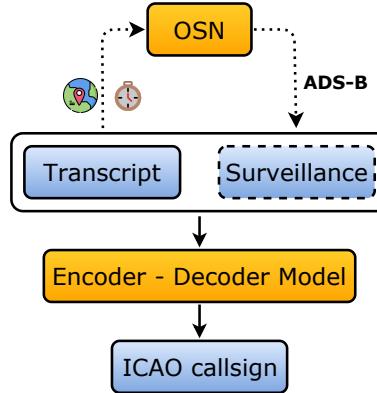


Figure 8: Proposed callsign recognition and understanding system. The dotted path marks the optional surveillance retrieval via OSN with the aid of the transcripts timestamp and VHF receiver location. Taken from [69].

<sup>29</sup>However, a sequence of acoustic embeddings can also be used for SC, e.g., emotion classification in raw speech [72].

Table 7: Different performance metrics for the speaker role detection experiments. Metrics reported on *ATCO2-test-set corpus* with a `bert-base-uncased` model trained on the splits from Table 2. @P, @R and @F1, refers to precision, recall and F1-score, respectively. Numbers in **bold** refers to the top performance per column.

Training Corpus	ATCO			PILOT			AVG @F1
	@P	@R	@F1	@P	@R	@F1	
LDC-ATCC	0.87	0.73	0.79	0.70	0.86	0.77	0.78
UWB-ATCC	0.88	<b>0.83</b>	<b>0.86</b>	<b>0.80</b>	0.85	0.82	<b>0.84</b>
LDC-ATCC + UWB-ATCC	<b>0.92</b>	0.78	0.85	0.76	<b>0.91</b>	<b>0.83</b>	<b>0.84</b>

### 7.3.1 Experimental Setup

The SC experiments are carried out in a very related manner to NER (see Section 7.1.1). Specifically, we use the same model (`bert-base-uncased`), hyperparameters (e.g., number of epochs), optimizer, dropout rates and so on. However, here, we fine-tuned the model on the SC task rather than NER. We append a linear layer with a dimension of 4 (following the classes structure from Section 3.2 of [49]) on top of the last layer of the BERT model, i.e., a two-class classification model.

We employed LDC-ATCC<sup>30</sup> and UWB-ATCC<sup>31</sup> datasets (see Table 1) for fine-tuning and *ATCO2-test-set corpus* for testing. In LDC-ATCC and UWB-ATCC databases, speaker roles tags for each sample are marked in the original transcripts. And, we use ground truth ASR transcripts the evaluation. We create speaker-independent train/test splits based on the original databases. The split IDs for each subset are registered in the public GitHub repository of this paper.

### 7.3.2 Results

We report the baseline results for speaker role detection in Table 7. Differently from NER, we only used *ATCO2-test-set corpus* for evaluation. We trained three models using different training datasets. From Table 7 we can see that pilots’ communications are more challenging for our model in comparison to the ones from ATCOs. For instance, in the model fine-tuned with LDC-ATCC corpus, there is a two-point drop in F1-scores for pilots, i.e.,  $0.79 \rightarrow 0.77$  F1-score. Similar behavior is seen in the model fine-tuned with UWB-ATCC corpus, i.e., a four-point drop in F1-scores,  $0.86 \rightarrow 0.82$ . However, models trained on the later show more robustness for both classes in comparison to the one trained with LDC-ATCC.

We also investigated the performance benefit of combining both datasets. For this experiment, we only obtained one point increase for the pilot class, while one point decrease for the ATCO class, both in comparison to the model trained on UWB-ATCC only. It is important to keep in mind that *ATCO2-test-set corpus* is a completely unseen dataset throughout all the experiments. We are convinced that integrating a small in-domain development set could boost the performances.

## 8 Legal and privacy aspects for collection of ATC recordings

In order to safely distribute and make available the ATCO Corpus to the community, we took into account legal and ethical considerations as a prerequisite to distribute this content both, commercially and as open-source package. The main question we faced was to determine whether we could legally record and distribute ATC. To answer that question, we inquired into how legislation and regulations treat ATC [74].

<sup>30</sup>The Air Traffic Control Corpus (LDC-ATCC) corpus is public in: <https://catalog.ldc.upenn.edu/LDC94S14A>. It comprises recorded speech for use in the area of ASR for ATC. The audio data is composed of voice communication traffic between various controllers and pilots.

<sup>31</sup>The UWB-ATCC corpus is released by the University of West Bohemia, and it can be downloaded for free in: <https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0001-CCA1-0>. The corpus contains recordings of communication between ATCOs and pilots. The speech is manually transcribed and labeled with the speaker information, i.e., whether ATCO or pilot is speaking and when.

Our first hypothesis was that ATC would fall under the rules of Intellectual Property law which regulate how authors and companies can collect and use immaterial works such as recordings, where we believed ATC could fall into. To confirm this hypothesis, we aimed to find out whether ATC could be considered as material that could be protected by intellectual property legislation. Therefore, we performed a thorough study of the two major legal intellectual property systems of the United States and Europe. This study showed that due to the specific characteristics of ATC, such as phraseology and context, these conversations could not be protected as such as they do not meet the originality threshold for protection.

Then we moved on to another hypothesis. We thought that even if these conversations can not be protected as such, they might be protected as part of databases collected by either aircraft companies or Air Navigation Services Providers (ANSPs). As part of our investigations, we came into contact with some of these stakeholders, but none of them replied favorably to our requests. For example, the National Air Traffic Services which handles the airspace for the United Kingdom replied to us that they could not provide their recordings unless mandated by a Court order. Moreover, the United Kingdom is one of the few countries that expressly prohibit recording ATC communications, as its legislation strictly prohibits the use of unlicensed recording apparatus<sup>32</sup>.

Other countries have a more lenient policy towards access and recording of ATC. Indeed, during our research we found out that ATC recorded in the US could be accessed on demand by formulating a request to the Federal Aviation Administration (FAA, hereafter). This is made possible by the use of the Freedom of Information Act (FOIA) that compels US administrations to make available certain types of information collected by these administrations during their operations. In the specific case of the FAA, Freedom of information regulations states that radio and computer data can be obtained upon request as stated in Chapter 4 Section 4-8-2 of the Facility Operation and Administration Order<sup>33</sup>. Nevertheless, during our exchanges with the FAA we found out that requests for audio files needed to concern the last 45 days as required by Chapter 12, section 2 Article 12-2-2 of the same order and had to be specific to an airport in order to be processed adequately by the administration. Future work may include the drafting of such a request to confirm the reality of those conversations.

For the airports based in Europe, we based our collection process on the existence of the Open Data Directive, formerly known as the Public Sector Directive, whose goal is to allow reuse and redistribution of data collected by public services, as we found out, many ANSPs are either State-owned or run by state administration for obvious security reasons. Therefore, we made a request in France to access data collected by the administration in charge of Air Traffic Control. We based our request on the provisions of French Law allowing to request the access to data produced by this administration. Regarding ATC, our request went up to the Commission d'Accès aux Documents Administratifs (Commission for access to administrative documents). This Commission ruled that the Direction Générale de l'Aviation Civile (DGAC) did not have to fulfill our request for data since they could not differentiate between civilian and military aircraft and that the recordings could leave the identification of speakers. However, following an *a contrario* interpretation, it can be assumed that since our project focussed on civilian aircraft and that we ensured the anonymization of the conversations before making them available, we could pursue data collection.

This previous ruling raised our concerns regarding the compliance of the project especially with the regulations related to protection of personal data, especially the EU General Data Protection Regulation (GDPR). GDPR is the main text regulating the collection and distribution of databases containing personal data. In the case of ATC, the speech data contains voiceprints of the pilots and ATCOs, which can be used as a mean of identification through speaker identification techniques. Thus, further precautions should be adopted in order to be able to collect this type of data. However, GDPR allow the collection of speech data when made in relation to reasons of substantive public interest, which we found applicable in our case since the project is aimed at enhancing airspace security.

---

<sup>32</sup>Further information in the following url: <https://www.legislation.gov.uk/ukpga/2006/36/section/48>

<sup>33</sup>JO 7210.3CC—Facility Operation and Administration available at [https://www.faa.gov/air-traffic/publications/atpubs/foa\\_html](https://www.faa.gov/air-traffic/publications/atpubs/foa_html)

## 9 Conclusions and Future Work

This article introduces, the *ATCO2 corpus*, a set of three corpora for research on robust automatic speech recognition and natural language understanding of air traffic control communications. In ATCO2, we have successfully created and deployed an operating pipeline for collecting, pre-processing and automatically transcribing ATC audio data. During the data collection period, we mostly relied on a worldwide community of volunteers that acted as ‘data feeders’. Then, a community of ‘data annotators’ employed the SpokenData transcription platform to generate gold annotations of a small portion of the collected data. The platform is up and running, and it is reachable on <https://www.spokendata.com/atco2>.

The *ATCO2 corpus* is divided in *ATCO2-PL-set corpus* and *ATCO2-test-set corpus*. The former contains more than 5000 hours of automatically transcribed ATC speech data, spanning more than ten airports in different continents (Table 3). While the latter, *ATCO2-test-set corpus*, contains gold annotations of 4 hours of ATC speech. A subset called *ATCO2-test-set-1h corpus* is offered for free in <https://www.atco2.org/data>. To the authors’ knowledge, this is the first public release of a large-scale database for research in the area of air traffic control communications.

In addition, we also cover baselines (our source code for data preparation and to replicate the NLU baselines will be stored in the following public GitHub repository <https://github.com/idiap/atco2-corpus>) over three different key tasks in the area of ATC. The first one is related to robust ASR, while the next two are about to NLU of ATC communications.

(1) We demonstrated that training an ASR system solely on *ATCO2-PL-set corpus* reaches competitive WERs on both, public and private databases (see Table 5). This is important because *ATCO2-PL-set corpus* is purely composed of pseudo labels generated by ATCO2 project seed ASR system. This can be the starting point for many researchers and companies worldwide that would like to use our corpora for testing and training robust ASR systems for ATC.

(2) We demonstrated that as much as 3000 utterances are needed to train and evaluate a BERT-based Named Entity Recognition system for ATC communications. This system is capable of detecting callsigns, commands, and values from the textual inputs. This NLU task is of special interest to the ATC community because this high-level information can be used to assist ATCOs in order to reduce their overall workload.

(3) Similarly, we developed a simple yet efficient BERT-based module that performs speaker role detection from textual inputs.

Finally, we believe that the ‘lessons learned’ in ATCO2 project and its recipe for collecting and pre-transcribing large-scale audio databases can be easily transferred to other applications, where data scarcity is a latent problem.

## Acknowledgements

This paper introduces the *ATCO2 Corpus* derived from a joint contribution from Clean Sky 2 Joint Undertaking (JU) and EU-H2020.

The work was fully supported by Clean Sky 2 Joint Undertaking (JU) and EU-H2020, under Grant Agreement No. 864702—ATCO2 (Automatic collection and processing of voice data from air-traffic communications).

## References

- [1] International Civil Aviation Organization, “ICAO phraseology reference guide,” 2020.
- [2] Martin Kocour, Karel Veselý, Igor Szöke, Santosh Kesiraju, Juan Zuluaga-Gomez, Alexander Blatt, Amrutha Prasad, Iuliia Nigmatulina, Petr Motlíček, Dietrich Klakow, et al., “Automatic processing pipeline for collecting and annotating air-traffic voice communication data,” *Engineering Proceedings*, vol. 13, no. 1, pp. 8, 2021.
- [3] Juan Zuluaga-Gomez, Petr Motlicek, Qingran Zhan, Karel Veselý, and Rudolf Braun, “Automatic Speech Recognition Benchmark for Air-Traffic Communications,” in *proceedings of Interspeech 2020*, 2020, pp. 2297–2301.

- [4] Juan Zuluaga-Gomez, Karel Veselý, Alexander Blatt, Petr Motlicek, Dietrich Klakow, Allan Tart, Igor Szöke, Amrutha Prasad, Saeed Sarfjoo, Pavel Kolčárek, et al., “Automatic call sign detection: Matching air surveillance data with air traffic spoken communications,” in *Multidisciplinary Digital Publishing Institute Proceedings*, 2020, vol. 59, p. 14.
- [5] José Manuel Cordero, Manuel Dorado, and José Miguel de Pablo, “Automated speech recognition in ATC environment,” in *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems*, 2012, pp. 46–53.
- [6] Stephane Pigeon, Wade Shen, Aaron Lawson, and David A van Leeuwen, “Design and characterization of the non-native military air traffic communications database (nnMATC),” in *Eighth Annual Conference of the International Speech Communication Association*, 2007.
- [7] Konrad Hofbauer, Stefan Petrik, and Horst Hering, “The ATCOSIM corpus of non-prompted clean air traffic control speech,” in *LREC*, 2008.
- [8] Luboš Šmídl, Jan Švec, Daniel Tihelka, Jindřich Matoušek, Jan Romportl, and Pavel Ircing, “Air traffic control communication (ATCC) speech corpora and their use for ASR and TTS development,” *Language Resources and Evaluation*, vol. 53, no. 3, pp. 449–464, 2019.
- [9] John Godfrey, “The Air Traffic Control Corpus (ATC0) - LDC94S14A,” 1994.
- [10] J Ferreiros, JM Pardo, R De Córdoba, Javier Macias-Guarasa, JM Montero, F Fernández, Valentin Sama, G González, et al., “A speech interface for air traffic control terminals,” *Aerospace Science and Technology*, vol. 21, no. 1, pp. 7–15, 2012.
- [11] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber, “Common voice: A massively-multilingual speech corpus,” *arXiv preprint arXiv:1912.06670*, 2019.
- [12] John J Godfrey, Edward C Holliman, and Jane McDaniel, “SWITCHBOARD: Telephone speech corpus for research and development,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*. IEEE Computer Society, 1992, vol. 1, pp. 517–520.
- [13] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: an ASR corpus based on public domain audio books,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
- [14] Charles T Hemphill, John J Godfrey, and George R Doddington, “The ATIS spoken language systems pilot corpus,” in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.
- [15] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al., “Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces,” *arXiv preprint arXiv:1805.10190*, 2018.
- [16] Juan Zuluaga-Gomez, Amrutha Prasad, Iuliia Nigmatulina, Saeed Sarfjoo, Petr Motlicek, Matthias Kleinert, Hartmut Helmke, Oliver Ohneiser, and Qingran Zhan, “How Does Pre-trained Wav2Vec2.0 Perform on Domain Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications,” *IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar*, 2023.
- [17] Juan Zuluaga-Gomez, Seyyed Saeed Sarfjoo, Amrutha Prasad, Iuliia Nigmatulina, Petr Motlicek, Karel Ondre, Oliver Ohneiser, and Hartmut Helmke, “BERTraffic: BERT-based Joint Speaker Role and Speaker Change Detection for Air Traffic Control Communications,” *IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar*, 2023.
- [18] Oliver Ohneiser, Saeed Sarfjoo, Hartmut Helmke, Shruthi Shetty, Petr Motlicek, Matthias Kleinert, Heiko Ehr, and Šarūnas Murauskas, “Robust command recognition for lithuanian air traffic control tower utterances,” in *Interspeech*, 2021.
- [19] Bruno Beek, E Neuberg, and David Hodge, “An assessment of the technology of automatic speech recognition for military applications,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 4, pp. 310–322, 1977.
- [20] Cheryl J Hamel, David Kotick, and Mark Layton, “Microcomputer system integration for air control training,” Tech. Rep., Naval Training Systems Center, Orlando FL, 1989.

- [21] K Matrouf, JL Gauvain, F Neel, and J Mariani, “Adapting probability-transitions in DP matching processing for an oral task-oriented dialogue,” in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 1990, pp. 569–572.
- [22] Robert Tarakan, Keith Baldwin, and Nicholas Rozen, “An automated simulation pilot capability to support advanced air traffic controller training,” in *The 26th Congress of ICAS and 8th AIAA ATIO*, 2008.
- [23] Hartmut Helmke, Oliver Ohneiser, Thorsten Mühlhausen, and Matthias Wies, “Reducing controller workload with automatic speech recognition,” in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. IEEE, 2016, pp. 1–10.
- [24] Hartmut Helmke, Oliver Ohneiser, Jörg Buxbaum, and Chr Kern, “Increasing ATM efficiency with assistant based speech recognition,” in *Proc. of the 13th USA/Europe Air Traffic Management Research and Development Seminar, Seattle, USA*, 2017.
- [25] Matthias Kleinert, Hartmut Helmke, Shruthi Shetty, Oliver Ohneiser, Heiko Ehr, Amrutha Prasad, Petr Motlicek, and Julia Harfmann, “Automated interpretation of air traffic control communication: The journey from spoken words to a deeper understanding of the meaning,” in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021, pp. 1–9.
- [26] Iain McCowan, Jean Carletta, Wessel Kraaij, Simone Ashby, Sébastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, et al., “The AMI meeting corpus,” in *Proceedings of the 5th international conference on methods and techniques in behavioral research*. Citeseer, 2005, vol. 88, p. 100.
- [27] Anthony Rousseau, Paul Deléglise, and Yannick Esteve, “TED-LIUM: an Automatic Speech Recognition dedicated corpus,” in *LREC*, 2012, pp. 125–129.
- [28] Matthias Kleinert, Hartmut Helmke, Gerald Siol, Heiko Ehr, Aneta Cerna, Christian Kern, Dietrich Klakow, Petr Motlicek, Youssef Oualil, Mittul Singh, et al., “Semi-supervised adaptation of assistant based speech recognition models for different approach areas,” in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. IEEE, 2018, pp. 1–10.
- [29] Ajay Srinivasamurthy, Petr Motlicek, Ivan Himawan, Gyorgy Szaszak, Youssef Oualil, and Hartmut Helmke, “Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control,” in *Proc. of the 18th Annual Conference of the International Speech Communication Association*, 2017.
- [30] Estelle Delpech, Marion Laignelet, Christophe Pimm, Céline Raynal, Michal Trzos, Alexandre Arnold, and Dominique Pronto, “A Real-life, French-accented Corpus of Air Traffic Control Communications,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- [31] L Graglia, B Favennec, and A Arnoux, “Vocalise: Assessing the impact of data link technology on the R/T channel,” in *24th Digital Avionics Systems Conference*. IEEE, 2005, vol. 1, pp. 5–C.
- [32] Stéphanie Lopez, Anne Condamines, Amélie Josselin-Leray, Mike O’Donoghue, and Rupert Salmon, “Linguistic analysis of english phraseology and plain language in air-ground communication,” *Journal of Air Transport Studies*, vol. 4, no. 1, pp. 44–60, 2013.
- [33] JC Segura, T Ehrette, A Potamianos, D Fohr, I Illina, PA Breton, V Clot, R Gemello, M Matassoni, and P Maragos, “The HIWIRE database, a noisy and non-native English speech corpus for cockpit communication,” *Online. <http://www.hewire.org>*, 2007.
- [34] Igor Szöke, Santosh Kesiraju, Ondřej Novotný, Martin Kocour, Karel Veselý, and Jan Černocký, “Detecting English Speech in the Air Traffic Control Voice Communication,” in *Proc. Interspeech 2021*, 2021, pp. 3286–3290.
- [35] Hartmut Helmke, Michael Slotty, Michael Poiger, Damián Ferrer Herrer, Oliver Ohneiser, Nathan Vink, Aneta Cerna, Petri Hartikainen, Billy Josefsson, David Langr, et al., “Ontology for transcription of ATC speech commands of SESAR 2020 solution PJ. 16-04,” in *IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. IEEE, 2018, pp. 1–10.
- [36] Chanwoo Kim and Richard M Stern, “Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis,” in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

- [37] Oldrich Plchot, Pavel Matejka, Ondrej Novotný, Sandro Cumani, Alicia Lozano-Diez, Josef Slavicek, Mireia Diez, Frantisek Grézl, Ondrej Glembek, Mounika Kamsali, et al., “Analysis of BUT-PT Submission for NIST LRE 2017,” in *Odyssey*, 2018, pp. 47–53.
- [38] Federico Landini, Ján Profant, Mireia Diez, and Lukáš Burget, “Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: theory, implementation and analysis on standard tasks,” *Computer Speech & Language*, vol. 71, pp. 101254, 2022.
- [39] Luboš Šmíd, Jan Švec, Daniel Tihelka, Jindřich Matoušek, Jan Romportl, and Pavel Ircing, “Air traffic control communication (ATCC) speech corpora and their use for ASR and TTS development,” *Language Resources and Evaluation*, vol. 53, no. 3, pp. 449–464, 2019.
- [40] Thomas Pellegrini, Jérôme Farinas, Estelle Delpech, and François Lancelot, “The Airbus Air Traffic Control speech recognition 2018 challenge: towards ATC automatic transcription and call sign detection,” *arXiv preprint arXiv:1810.12614*, 2018.
- [41] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., “The Kaldi speech recognition toolkit,” in *IEEE workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number CONF.
- [42] Nelson Morgan, Herve Bourlard, Steve Renals, Michael Cohen, and Horacio Franco, “Hybrid neural network/hidden markov model systems for continuous speech recognition,” in *Advances in Pattern Recognition Systems Using Neural Network Technologies*, pp. 255–272. World Scientific, 1993.
- [43] Herve A Bourlard and Nelson Morgan, *Connectionist speech recognition: a hybrid approach*, vol. 247, Springer Science & Business Media, 1993.
- [44] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, “Speaker adaptation of neural network acoustic models using i-vectors,” in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 55–59.
- [45] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, “Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI,” in *INTERSPEECH 2016, San Francisco, CA, USA, September 2016*. 2016, pp. 2751–2755, ISCA.
- [46] Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur, “Semi-orthogonal low-rank matrix factorization for Deep Neural Networks,” in *Proceedings of INTERSPEECH 2018*, 09 2018, pp. 3743–3747.
- [47] Juan Zuluaga-Gomez, Iuliia Nigmatulina, Amrutha Prasad, Petr Motlicek, Karel Veselý, Martin Kocour, and Igor Szöke, “Contextual Semi-Supervised Learning: An Approach to Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems,” in *Interspeech*, 2021, pp. 3296–3300.
- [48] Martin Kocour, Karel Veselý, Alexander Blatt, Juan Zuluaga Gomez, Igor Szöke, Jan Cernocky, Dietrich Klakow, and Petr Motlicek, “Boosting of Contextual Information in ASR for Air-Traffic Call-Sign Recognition,” in *Interspeech*, 2021, pp. 3301–3305.
- [49] Iuliia Nigmatulina, Juan Zuluaga-Gomez, Amrutha Prasad, Seyyed Saeed Sarfjoo, and Petr Motlicek, “A two-step approach to leverage contextual data: speech recognition in air-traffic communications,” in *ICASSP*, 2022.
- [50] Iuliia Nigmatulina, Rudolf Braun, Juan Zuluaga-Gomez, and Petr Motlicek, “Improving callsign recognition with air-surveillance data in air-traffic communication,” *arXiv preprint arXiv:2108.12156*, 2021.
- [51] Yi Lin, “Spoken instruction understanding in air traffic control: Challenge, technique, and application,” *Aerospace*, vol. 8, no. 3, pp. 65, 2021.
- [52] Amrutha Prasad, Juan Zuluaga-Gomez, Petr Motlicek, Oliver Ohneiser, Hartmut Helmke, Saeed Sarfjoo, and Iuliia Nigmatulina, “Grammar Based Identification Of Speaker Role For Improving ATCO And Pilot ASR,” *arXiv preprint arXiv:2108.12175*, 2021.
- [53] Hartmut Helmke, Matthias Kleinert, Shruthi Shetty, Oliver Ohneiser, Heiko Ehr, Hörður Arilíusson, Teodor S Simiganoschi, Amrutha Prasad, Petr Motlicek, Karel Veselý, et al., “Readback error detection by automatic speech recognition to increase ATM safety,” in *Proceedings*

*of the Fourteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2021), Virtual Event*, 2021, pp. 20–23.

- [54] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [55] Sowmya Vajjala and Ramya Balasubramaniam, “What do we Really Know about State of the Art NER?,” *arXiv preprint arXiv:2205.00034*, 2022.
- [56] Ralph Grishman and Beth M Sundheim, “Message understanding conference-6: A brief history,” in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.
- [57] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, “Natural language processing (almost) from scratch,” *Journal of machine learning research*, vol. 12, pp. 2493–2537, 2011.
- [58] Jakub Piskorski, Lidia Pivovarova, Jan Šnajder, Josef Steinberger, and Roman Yangarber, “The first cross-lingual challenge on recognition, normalization, and matching of named entities in Slavic languages,” in *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, 2017, pp. 76–85.
- [59] Vikas Yadav and Steven Bethard, “A survey on recent advances in named entity recognition from deep learning models,” in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2145–2158.
- [60] Yinhai Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [61] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen, “DeBERTa: Decoding-enhanced BERT with Disentangled Attention,” in *International Conference on Learning Representations*, 2021.
- [62] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” *arXiv preprint arXiv:1910.01108*, 2019.
- [63] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pieric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al., “Transformers: State-of-the-art natural language processing,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.
- [64] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, et al., “Datasets: A community library for natural language processing,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2021, pp. 175–184.
- [65] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [66] Dan Hendrycks and Kevin Gimpel, “Gaussian error linear units (gelus),” *arXiv preprint arXiv:1606.08415*, 2016.
- [67] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, “On the difficulty of training recurrent neural networks,” in *International conference on machine learning*. PMLR, 2013, pp. 1310–1318.
- [68] Ilya Loshchilov and Frank Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations*, 2019.
- [69] Alexander Blatt, Martin Kocour, Karel Veselý, Igor Szőke, and Dietrich Klakow, “Call-sign recognition and understanding for noisy air-traffic transcripts using surveillance information,” in *ICASSP*, 2022, pp. 8357–8361.
- [70] Zhiyong He, Zanbo Wang, Wei Wei, Shanshan Feng, Xianling Mao, and Sheng Jiang, “A Survey on Recent Advances in Sequence Labeling from Deep Learning Models,” *arXiv preprint arXiv:2011.06727*, 2020.

- [71] Cheng Zhou, Boris Cule, and Bart Goethals, “Pattern based sequence classification,” *IEEE Transactions on knowledge and Data Engineering*, vol. 28, no. 5, pp. 1285–1298, 2015.
- [72] Tilak Purohit, Imen Ben Mahmoud, Bogdan Vlasenko, and Mathew Magimai Doss, “Comparing supervised and self-supervised embedding for ExVo Multi-Task learning track,” *arXiv preprint arXiv:2206.11968*, 2022.
- [73] Marouane Birjali, Mohammed Kasri, and Abderrahim Beni-Hssane, “A comprehensive survey on sentiment analysis: Approaches, challenges and trends,” *Knowledge-Based Systems*, vol. 226, pp. 107134, 2021.
- [74] Mickaël Rigault, Claudia Cevenini, Khalid Choukri, Martin Kocour, Karel Veselý, Igor Szoke, Petr Motlicek, Juan Pablo Zuluaga-Gomez, et al., “Legal and ethical challenges in recording air traffic control speech,” in *Proceedings of the Workshop on Ethical and Legal Issues in Human Language Technologies and Multilingual De-Identification of Sensitive Data In Language Resources within the 13th Language Resources and Evaluation Conference*, 2022, pp. 79–83.

## A Automatic Transcription Engine

This appendix describes in details how we collected the audio and metadata that brought to live the *ATCO2 corpus*. We mainly rely on the automatic transcription engine, described in more details in Section 4.3. The automatic transcription engine is implemented as a scalable cloud service. It communicates with other services (or partners) using APIs. This service is designed to process large flows of data produced by data feeders.<sup>34</sup>

The data is pushed to this service by OSN<sup>35</sup> servers by calling an API request and providing a job setting JSON file. After the request is accepted, settings parameters are processed and the job is stored in an internal queue for processing. The user (in this case, OSN) may have an ability to tweak the settings and to affect the processing pipeline and the result. Namely:

- Audio input format choices;
- Rejection threshold for too long audios;
- Rejection threshold for too short audios;
- Rejection threshold for too noisy audios;
- Rejection threshold for non-English audios;
- Switching the language of automatic speech recognizer.

Most of these are actually disabled due to security reasons (not to interrupt the processing pipeline), but may be easily enabled on the fly if needed. The overall data flow model is described in Figure 9. Any new job (request for a full automatic annotation of recording) accepted via API on the SpokenData<sup>36</sup> side is processed by a master processing node. The job is enqueued into a workload manager queue. Once there is a free processing slot, the job is submitted to a processing server, or worker. The master processing node then informs the OSN server about the state of the job by calling a callback.

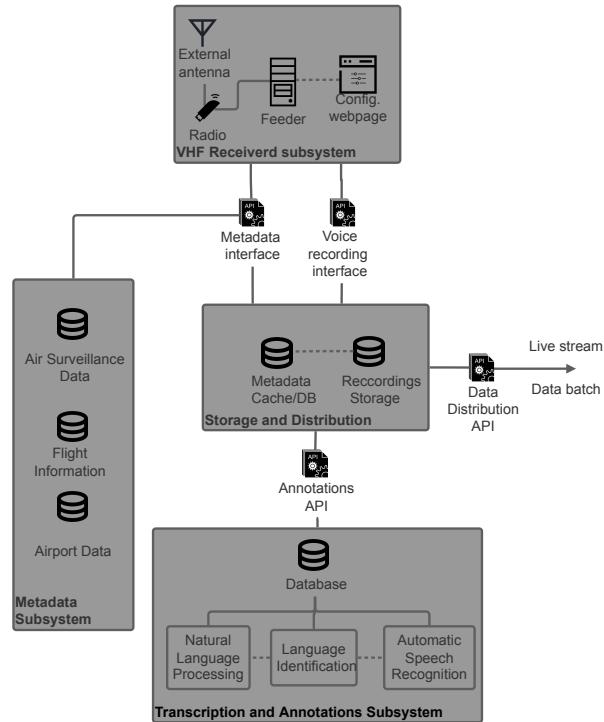


Figure 9: Overall ATCO2 communication schema.

<sup>34</sup>Enthusiasts that act as ‘feeders’ of ATC speech and contextual ATC data (surveillance). See Section 4.1.

<sup>35</sup>OpenSky Network: <https://opensky-network.org/>.

<sup>36</sup>Industrial partner: <https://www.spokendata.com/atco2>.

## B Full processing pipeline

The processing pipeline is implemented as a Python script which follows a configuration file, i.e., `worker.py`. The configuration file allows us to modify the logic and flow of the data in the pipeline on the fly. It allows parallelism, forking, and conditions. In principle, `worker.py` consists of global definitions (constants), blocks (local definitions) and links (an acyclic oriented graph) between blocks. The processing pipeline is given on Figure 10. For instance, we address in previous work [2, 4] early implementations of each technology, e.g., segmentation and diarization, ASR or named entity recognition. All the technologies and tools are encapsulated in a BASH scripts with a unified interface.

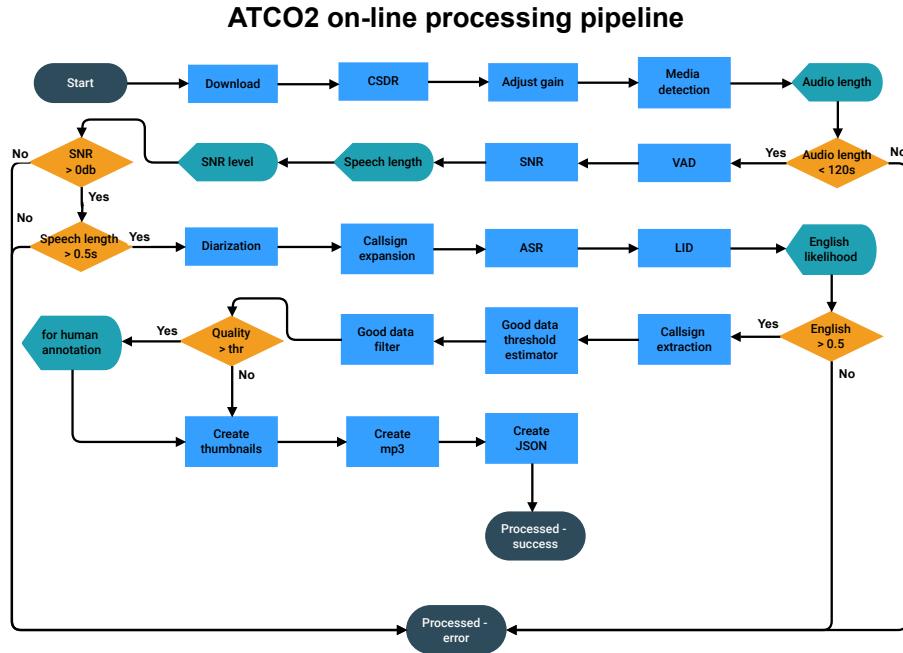


Figure 10: Diagram of the processing pipeline. The blue rectangles are processes. The cyan arrow blocks are internal callback events, where the pipeline informs the master node about progress and sends intermediate results. Finally, the orange rhombuses are conditions. Here the intermediate results are taken (e.g., an SNR level) and it is decided if it makes sense to continue (clean audio) or stop processing, e.g., a given segment reached the maximum allowed noise level. A final internal callback is run when the pipeline finishes. It triggers the API to call the OSN server with the particular callback (processing done as *OK* or *ERROR*).

The first row of blocks from Figure 10 refers to segmentation and demodulation. First, an antenna, and a recording device jointly capture the radio signal, which is divided into segments containing portions where the transmission was “active”, the silent parts are not recorded (push-to-talk is used in ATC voice communication). This functionality is part of the RTLSDR-Airband audio recording software, from which we dump the raw I/Q signal. Second, we convert this complex I/Q radio signal into a waveform signal by a software defined radio CSDR. The first part is done in the recording device, while the second is done at OpenSky Network servers. Next, we do “signal-to-noise ratio (SNR) filtering” (second row), the purpose is to remove the recordings that are too noisy. In bad recording conditions, we can end up in a situation in which the voice is not intelligible. The following step is “diarization” (third row). In the automatically segmented data, some recordings contain more than one speaker. This is a problem, because we intend to automatically transcribe speaker turns of single speakers. And, for subsequent NLP/SLU tasks, it is important to separate the speaker turns as well. The diarization solves this by splitting the audio into segments with single-speakers and assigning them speaker labels. In the ASR step, we simply convert “speech-to-text”. This is done by our ASR system that we build with tools from the Kaldi toolkit [41]. The output from this step are transcripts which, inevitably, contain some errors. To improve the accuracy of the transcripts,

we use contextual information (call-sign lists from surveillance data). The mechanism is that we give score discounts to some rare ‘words’ in the lattice-generation step, or we give score discounts to some ‘word sequences’ by rescoring lattices of alternative hypotheses (further details in [48, 50]. The call-sign lists come from the traffic monitoring databases of OpenSky Network. Next, the transcripts are used as an input for the English language detection (ELD) system. The purpose is to be able to discard non-English audio data. The typical state-of-the-art language identification system is based on acoustic modeling, and uses audio as input. However, for the ATC speech, we don’t need to “identify” the non-English language, so we developed a “lexical English detection system” which uses transcripts and confidence scores produced by ASR as its inputs. For ATC speech, this worked better than the “traditional” acoustic language identification method. The last automatic operation is “post-processing by NLP”. Currently, we have a Callsign-code Extractor that returns the callsign in ICAO format like “DLH77RM” belonging to an aircraft. Finally, some processed data go through “human correction”, and some data are kept with the automatic labels. The former case produced *ATCO2-test-set corpus*, while the latter, *ATCO2-PL-set corpus*.

## C Unification of transcripts

This appendix conveys our main results of transcripts unification and lexicon formatting. Note that the description below is related to the databases employed to train the seed ASR engine used during the pre-transcription process, described in Section 4.3. Special attention was devoted to the unification of words from the radiotelephony alphabet and numbers (see *ICAO alphabet* column from Table 8). Note that we map the word 'niner' → 'nine', and in the pronunciation lexicon, we allow the word 'nine' to be pronounced as 'n ay1 n er0' (phoneme-based format). Also, some standard expressions can be written as two words or as a single word. For some of the frequent ones, we selected one version that is used systematically (see *Common expressions* column from Table 8). We also rectify some airline designators that are part of the callsigns uttered by the ATCOs and pilots (see *Airline designator* column from Table 8).

We derive a table of mapping rules by extracting insights from a diverse list of airline designators. In total, we have a list of 5.4k airline designators, out of these, there are 1.8k multi-word airline designators. The airline designators ligatured by underscore are easier to be produced by the 'speech-to-text' system as the tokens are longer, and there is also less uncertainty to be modelled by the language model. Finally, we pay extra attention to the transcripts generated by the ATCO2 community of volunteers. Like in any other human input, there might be typos or other types of transcription errors. It is necessary to at least revise the transcripts by the 2nd round of human inspection, where the errors are ideally fixed.

Table 8: Normalization rules applied for annotation.

Unification of transcripts		
ICAO alphabet	Common expressions	Airline designators
alpha → alfa	take off → takeoff	qatar → qatari
charly → charlie	call sign → callsign	turkey → turkish
juliet → juliett	readback → read back	air france → airfrans
oskar → oscar	flightlevel → flight level	norshuttle → nor shuttle
xray → x-ray	stand by → standby	airvan → air van
zoulou → zulu	start up → startup	rynair → ryanair
whisky → whiskey	goodbye → good bye	airbaltic → air_baltic
tripple → triple	clear for → cleared for	air berlin → air_berlin
niner → nine	lineup → line up	air canada → air_canada
0 → zero	clear for → cleared for	air china → air_china
1 → one	turnright → turn right	air europe → air_europe
2 → two	oclock → o'clock	jet stream → jet_stream
3 → three	o clock → o'clock	jetstream → jet_stream
4 → four	push back → pushback	k 1 m → k_1_m
5 → five	descent direct → descend direct	klm → k_1_m
6 → six	goodbye → good bye	korean air → korean_air
7 → seven	goodday → good day	koreanair → korean_air
8 → eight	turbulence → turbulence	wizzair → wizz_air
9 → nine	til → till	top_jet → topjet

## D How a Sample From ATCO2 corpora Looks Like?

Example of human annotations for a recording of *ATCO2-test-set corpus* in XML format. This file encodes most of the metadata. If more than one segment is present, it means there are two or more in the recording:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <data>
3     <segment>
4         <start>0</start>
5         <end>2.93</end>
6         <speaker>B</speaker>
7         <speaker_label>pilot</speaker_label>
8         <text>[unk] [#callsign] Quebec Lima[/#callsign] [# command] confirm cleared for ILS[/#command] [unk]</text>
9         <tags>
10            <correct>0</correct>
11            <correct_transcript>1</correct_transcript>
12            <correct_tagging>0</correct_tagging>
13            <non_english>0</non_english>
14        </tags>
15    </segment>
16    <segment>
17        <start>2.99</start>
18        <end>10.45</end>
19        <speaker>A</speaker>
20        <speaker_label>ATCO approach</speaker_label>
21        <text>[unk] [#callsign] Quebec Lima[/#callsign] [# command] affirm cleared ILS approach[/#command] [#value] runway one four[/#value] [#command] if you go around[/#command] [#value] runway one four[/#value] [#command] report in localizer established[/# command]</text>
22        <tags>
23            <correct>0</correct>
24            <correct_transcript>1</correct_transcript>
25            <correct_tagging>0</correct_tagging>
26            <non_english>0</non_english>
27        </tags>
28    </segment>
29 </data>
```

Listing 1: XML tagged example from *ATCO2-test-set corpus*. This example contains two recordings separated by the `<segment>` tag.

Basic details from the previous XML tagged segment:

- `<segment> ... </segment>`: one sample of data. One recording may have one or more segments;
- `<start> ... </end>`: timing information with speech activity by the speakers;
- `<speaker> ... </speaker>`: speaker information to identify whether the segment is from an ATCO or pilot. Unknown cases are tagged with `<UNK>`
- `<text> ... </text>`: ground truth transcripts with high-level entities annotations (callsigns, commands and values). Not all the segments contains these annotations.
- `<tags> ... </tags>`: extra metadata.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/39435558>

# A corpus-based approach to generalising a chatbot system

Thesis · January 2005

Source: OAI

---

CITATIONS

15

READS

1,078

2 authors:



Bayan Abu Shawar

Al Ain University

60 PUBLICATIONS 1,274 CITATIONS

[SEE PROFILE](#)



Eric Atwell

University of Leeds

408 PUBLICATIONS 4,448 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Arabic Dialects Text Classification [View project](#)



Research [View project](#)

# A Corpus-based approach to generalising a chatbot system

**Bayan Abu Shawar**

University of Leeds

Leeds LS2 9JT, England

[bshawar@comp.leeds.ac.uk](mailto:bshawar@comp.leeds.ac.uk)

**Eric Atwell**

University of Leeds

Leeds LS2 9JT, England

[eric@comp.leeds.ac.uk](mailto:eric@comp.leeds.ac.uk)

**Abstract:** International research in NLP is dominated by work on English. NLP techniques and systems can be ported to other natural languages, but this is generally a labour-intensive task, requiring scarce computational and linguistic expertise; hence minority languages are poorly represented in NLP technology. We present an automated approach to porting an NLP technology, the AIML-based chatbot, to new languages, by using a corpus in the target language to retrain the chatbot. We have successfully automated production of chatbots talking French, and Afrikaans; and are developing further demonstrators in Spanish and Arabic.

**Keywords:** chatbot, dialogue, corpus, machine learning, English, French, Afrikaans, Arabic

Human machine conversation is a new technology to facilitate communication between users and computers via natural language. A chatbot is a conversational agent that interacts with users turn by turn using natural language. ALICE (<http://www.alicebot.org/>, Abu Shawar and Atwell 2002) is a chatbot system that implements various human dialogues, using AIML (Artificial Intelligent Markup Language), a version of XML, to represent the patterns and templates underlying these dialogues. The basic units of AIML objects are categories. Each category is a rule for matching an input and converting to an output, and consists of a pattern, which represents the user input, and a template, which implies the ALICE robot answer. The AIML pattern is simple, consisting only of words, spaces, and the wildcard symbols \_ and \*. The words may consist of letters and numerals, but no other characters. Words are separated by a single space, and the wildcard characters function like words. The pattern language is case invariant.

Since the primary goal of chatbots is to mimic real human conversations, we developed a java program that learns from dialogue corpora to generate AIML files in order to modify ALICE to behave like the corpus. Two versions of the program were generated:

The first version is based on simple pattern template category, so the first turn of the speech is the pattern to be matched with the user input, and the second is the template that holds the

robot answer. This version was tested using the English-language Dialogue Diversity Corpus (DDC), see <http://www-rcf.usc.edu/~billmann>, (Abu Shawar and Atwell 2003) to investigate the problems of utilising dialogue corpora. The DDC is a collection of links to different dialogue corpuses in different fields. These annotated texts are transcribed from recorded dialogues between more than two speakers. The dialogue corpura contain linguistic annotation that appears during the spoken conversation such as overlapping, and using some linguistic fillers. To handle the linguistic annotations and fillers, the program is composed of fours phases as follows:

1. Phase One: Read the dialogue text from the corpus and insert it in a vector.
2. Phase Two: Text reprocessing modules, where all linguistic annotations such as overlapping, fillers and other linguistic annotations are filtered.
3. Phase Three: converter module, where the pre-processed text is passed to the converter to consider the first turn as a pattern and the second as a template. Removing all punctuation from the patterns and converting it to upper case is done during this phase.
4. Phase Four: Copy these atomic categories in an AIML file.

The most significant problem with the DDC is the unstructured annotations used within its files. We applied the same program to a French dialogue corpus (Kerr 1983), which also

required changing the pre processing text since it has its own specific annotations.

The second version of the program has a more general approach to finding the best match against user input from the learned dialogue. At first we decided to treat the problem of having more than two speakers within the dialogue corpora by 'recycling' each turn to be a pattern on one category and a template on the consecutive one. We used the same modules generated in the first version in order to read and pre-process the text. A restructuring module was added to evolve the program. The restructuring module searched the pattern template vector, to map all patterns with the same response to one form, and to transfer all repeated pattern with different templates to one pattern with a random list of different responses. We then used an Afrikaans corpus (Van Rooy, 2002) to generate two versions of ALICE: Afrikaana speaks only Afrikaans, and AVRA is bilingual and speaks both English and Afrikaans (most Afrikaans speakers are in fact bilingual). The bilingual version combined the standard ALICE AIML files that are written in English and the Afrikaana AIML file that is written just in Afrikaans. We used the <http://www.pandorabots.com/pandora> web-hosting service to make our chatbots available for use over the World Wide Web. User feedback from Afrikaans speakers suggested that we needed to extend the pattern-matching to enhance the responses generated.

To do this, we used the first word approach, based on the generalisation that the first word of an utterance may be a good clue to an appropriate response: if we cannot match the whole input utterance, then at least we can try matching just the first word. For each atomic pattern, we generated a default version that holds the first word followed by wildcard to match any text, and then associated it with the same atomic template. Unfortunately this approach still failed to satisfy our trial users, so we decided to use the most significant approach to augment the first word approach.

Instead of assuming the first word of an utterance is most "significant", we look for the word in the utterance with the highest "information content", the word that is most specific to this utterance compared to other utterances in the corpus. This should be the

word that has the lowest frequency in the rest of the corpus. We choose the most significant approach to generate the default categories, because usually in human dialogues the intent of the speakers is hiding in the least-frequent, highest-information word. We extracted a local least frequent list from the Afrikaans corpus, and then compared it with each token in the pattern to specify the most significant word within that pattern. Four categories holding the most significant word were added to handle the positions of this word first, middle, last or alone. The feedback showed improvement in user satisfaction.

To avoid the problems raised using corpus-based approach, the ideal training corpus must have the following characteristics: two speakers, structured format, short, obvious turns without overlapping, and without any unnecessary notes, expressions or other symbols that are not used when writing a text.

Even such "idealised" transcripts may still lead to a chatbot which does not seem entirely "natural": although we aim to mimic the natural conversation between humans, the chatbot is constrained to chatting via typing, and the way we write is different from the way we speak.

Building French and Afrikaans versions of ALICE demonstrated the general approach. We propose to demonstrate the program further by developing other versions, including Spanish and Arabic chatbots.

## References

- Abu Shawar B, Atwell E. 2002. A comparison between ALICE and Elizabeth Chatbot systems, Technical report, School of Computing, University of Leeds.
- Abu Shawar B, Atwell E. 2003. Using dialogue corpora to retrain a chatbot system, Proceedings of Corpus Linguistics 2003, pp681-690, Lancaster University.
- Kerr, B. 1983. Minnesota Corpus. University of Minnesota Graduate School, Minneapolis.
- Van Rooy, B. 2002. Transkripsiehandleiding van die Korpus Gesproke Afrikaans. [Transcription Manual of the Corpus Spoken Afrikaans.] Potchefstroom University.



# An AI toolkit for libraries

Now that artificial intelligence (AI) tools are being widely used across academic publishing, how can we make informed assessments of these utilities? There is a need for a set of skills for evaluating new tools and measuring existing ones, which should enable anyone commissioning or managing AI utilities to understand what questions to ask, what parameters to measure and possible pitfalls to avoid when introducing a new utility. The skills required are not technical. Potential problems include bias in the corpus, a poor training set or poor use of metrics for evaluation. This article gives a quick overview of some of areas where AI tools are being used and how they work. It then provides a checklist for assessment. The goal is not to discredit AI, but to make effective use of it.

## Keywords

AI; NLP; evaluation; metrics; research support

## Introduction

A colleague walks up excitedly to you. 'I've just discovered a really cool AI app to speed up the submissions process for my articles – it does what we currently do by hand twice as quick, and all you have to do is to press a few buttons! Check it out!' How should you respond? The interface certainly looks well designed and appealing. There is not much information on the site, but the developers seem to have thought of everything. Do you feel qualified to give an opinion on this tool?

The aim of this article is to outline a framework for evaluating artificial intelligence- (AI-) based tools, without the need to have or to acquire detailed technical knowledge of how they were developed or any requirement to understand computing languages such as Python, or indeed any advanced maths. Nonetheless, the criteria for evaluation described here are crucial for the successful use of AI. The article argues that the human users may in some cases be better placed to evaluate the capabilities of a tool than the original developers, who quite possibly were not in a position to appreciate the context in which it would be used.

AI tools can possibly provide a way to reduce the time taken to discover or to submit academic content; they have the potential to improve the quality of published articles by running more detailed and more accurate checks in advance of publication. However, it is not the intention of this article to provide arguments for or against the use of AI tools compared with human evaluation. Instead, the aim is to outline how such tools can be assessed in a real-world setting. There are already many tools making use of AI in our daily lives, but we don't always realize that AI is involved. For many of these tools, for example some of the components of the Google search engine, their introduction was without debate, and we have subsequently become accustomed to their strengths and weaknesses.

Like many new technologies, AI has been viewed through widely different perspectives during its long lifetime, dating back over 75 years to the 1950s<sup>1</sup> – from wild optimism to being written off. Unfortunately, both attitudes are wide of the mark, and neither extreme



MICHAEL UP SHALL

Consultant

'a framework for evaluating artificial intelligence- (AI-) based tools'

<sup>2</sup> is helpful for a balanced appraisal of AI tools. Repeatedly during AI's lifetime, many highly regarded thinkers have described AI either with glowing optimism, or expected AI to bring about disaster:

'Artificial intelligence will reach human levels by around 2029.' Ray Kurzweil.<sup>2</sup>

'The development of full artificial intelligence could spell the end of the human race.' Stephen Hawking, BBC Interview 2014.<sup>3</sup>

'At some stage therefore, we should have to expect the machines to take control.'  
Alan Turing, draft of lecture, 1951.<sup>4</sup>

'The internet makes us superficial.' Nicholas Carr, *The Shallows*, a 2010 book that has been cited 265 times.<sup>5</sup>

'With artificial intelligence we are summoning the demon.' Elon Musk.<sup>6</sup>

Perhaps more alarmingly, because they write from a perspective of widespread use of AI, several practitioners experienced in AI have recently written books that stress the negative implications of AI. Kate Crawford in her *Atlas of AI*<sup>7</sup> wrote about how AI creates low-paid work. Cathy O'Neill described the effects of AI bias on school selection panels in her *Weapons of Math Destruction*.<sup>8</sup> Erik Larson complains about the overextension of AI in an unthinking way in his *The Myth of Artificial Intelligence*.<sup>9</sup> You could be forgiven for thinking from these books that AI is an unmitigated disaster. Even the triumphs seem to turn out to be heavily qualified: we have been promised self-driving cars for many years but, apart from a relatively small number of controlled trials, self-driving cars have still not become commonplace. Perhaps this is simply technology evolving faster than we think in the long term but more slowly in the short term, or perhaps this is an inherent limitation of AI.

In the light of these warnings, should we be implementing AI tools or calling for more investigation? Should we abandon AI, or use it intelligently? The suggested solution here is to concentrate on a small subset of available AI tools and to follow a clear methodology for assessing them.

'Should we abandon AI, or use it intelligently?'

## What AI tools consist of

The narrow, or limited, AI described here is based around a few components. Present-day AI for text tools for academic purposes typically comprises:

- a corpus
- a training set
- a test set
- an algorithm.

The 'corpus' is the body of content that you wish to analyse, for example, all scientific research articles published in the last 20 years. The corpus contains some information or characteristic that you wish to extract. A corpus need not only be text – there are corpora for facial recognition, for example, as well as the often referred to collections of images of cats and dogs.

The 'training' set is a subset of the corpus, which has been tagged in some way to identify the characteristic you are looking for. Thus, a training set for cats and dogs might be 100 images that were tagged by a human as one or the other. Another example is the Modified National Institute of Standards and Technology (MNIST) database of handwritten numbers,<sup>10</sup> which shows many examples of the range of styles used when humans write numbers by hand, see Figure 1.

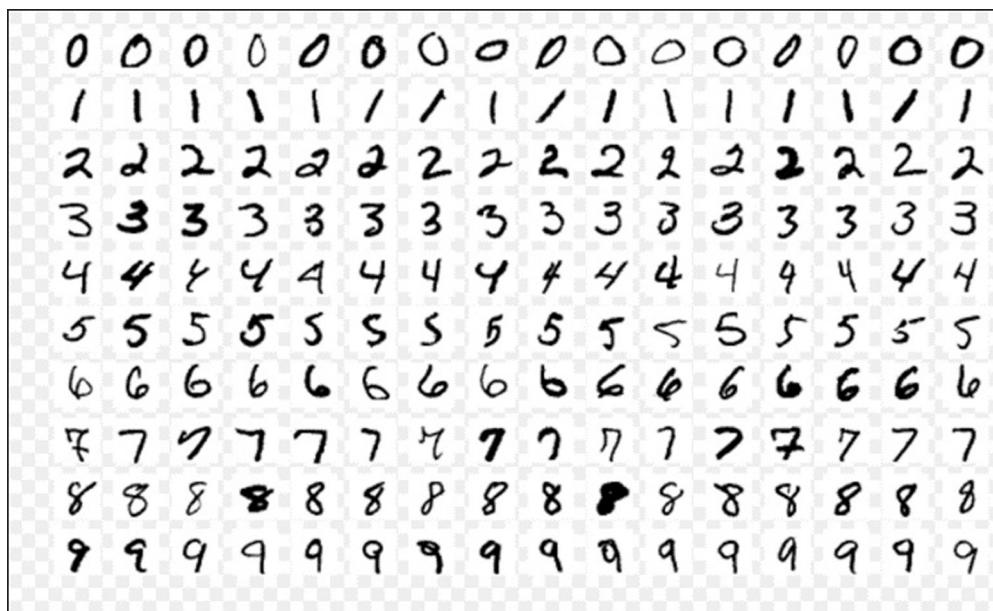


Figure 1. Example of the MNIST database of handwritten numbers

The ‘test set’ is the collection of documents to be used for trialling the algorithm, to see how successfully it carries out the operation. The ‘algorithm’ is simply the tool that looks at each item in the corpus and enables a decision to be made. An algorithm may be as simple (and frequently is as simple) as matching a pattern; so, for example, if you give the device ten handwritten examples of the numbers 0 to 9, then the machine is asked to find the closest match between the training set and the test set. A cookery recipe is an algorithm, as is a way of sorting documents – by date, or by subject. Much of computing is based around identifying the most effective algorithm to solve a specific problem, for example, how to sort a collection of numbers into numerical order.

Surveys suggest that the public has a low awareness that algorithms are being used.<sup>11</sup> Worse, there is a common misconception that when algorithms are used, they are the cause of any defects of the tool. One of the myths about present-day AI is that it is entirely about algorithms. If only algorithms were revealed in public, is the argument, then all the mysteries of AI would be revealed.<sup>12</sup> Reports in the media have tended to reinforce this misconception, accusing the algorithm of creating the problem; a 2019 study by the European Parliament on algorithmic accountability does not mention the term corpus.<sup>13</sup> Newspaper reports about using an algorithm to determine the results of public examination suggested the algorithm was the cause of the issue, rather than the (undocumented) way it had been implemented: ‘We all remember the A-levels fiasco, when an algorithm decided what the results should be ... the poorest students received worse marks’.<sup>14</sup>

**‘One of the myths about present-day AI is that it is entirely about algorithms’**

More exactly, the success or failure of AI is as much based on the corpus as it is on the algorithm. If the corpus used has an imbalance of gender, ethnic group or geographical origin, then the algorithm will simply replicate that bias.

To summarize, artificial general intelligence raises many issues that, to be honest, are of little relevance to most present-day AI, even though they will keep leading-edge researchers busy for years. Narrow AI tools can, if implemented sensibly, greatly enhance our ability to carry out many of the tasks in the academic workflow. How these tools are selected and implemented is all-important. How can these tools – and the corpora they are based on – be evaluated? The role of the library is crucial in providing guidance on real-world selection, implementation and, finally, appraisal and metrics.

## What is present-day AI?

For many years, AI researchers have been obsessed with creating AGI: artificial general intelligence. One algorithm could answer all the questions in the universe. The idea behind a universal algorithm is, in the words of AI researcher Pedro Domingos,

'If it exists, the Master Algorithm can derive all knowledge in the world—past, present, and future—from data. Inventing it would be one of the greatest advances in the history of science.'<sup>15</sup>

The idea of a universal general intelligence was widespread in the 1960s. It fell out of favour for several years, but traces of it are still evident today in research departments. According to Wikipedia,<sup>16</sup> there were 72 active AGI projects running in 2020, which indicates that many researchers continue to look for a unified solution via the use of AI, rather than making use of limited tools in specific contexts, which is what this article is concentrating on.

For the purpose of this article, the master algorithm will be ignored and the focus will be a smaller set of tools, typically employed for just one purpose. Formally, the tools described here make use of what is called 'supervised' or 'semi-supervised' machine learning.<sup>17</sup> Supervised means there is some human involvement in setting up the tool, usually in determining what the correct answers should be. 'Machine learning' (ML) means the use of a computer to follow a pattern, whether or not the pattern is identified by a human. 'Natural language processing' or NLP means the identification of patterns in spoken or written text.

## Do we know that AI is being used?

This is a more fundamental question than might be imagined. There are many examples of AI tools in use without any mention that AI is being used, although, increasingly, the impact of the AI tool might be too subtle to notice. For example, in a Google blog post about BERT,<sup>18</sup> an ML technique for NLP, the benefit shown was simply the ability to link a preposition with a noun. Whereas earlier search tools tended to ignore prepositions and just focus on nouns, this more sophisticated tool was able to handle a question about a traveller from Brazil to the USA. It identifies a meaningful connection between the 'to' and the 'USA'.

In social media and product literature, the term AI is frequently used as a buzzword to give the impression that a tool is more sophisticated than it really is. In practice, the kind of small-scale AI described above is very closely linked to 'string matching' or other well-established simple techniques. String matching means the use of a machine to identify instances of a sequence of characters in a text.<sup>19</sup> Eslami claims that once users are shown they are interacting with an algorithm, rather than with a human, they are reassured; there certainly appears to be widespread suspicion of an algorithm making decisions for a human. Not revealing that there is no human involved makes things much worse; the users feel cheated, because they were not told. Google search is an example where we as everyday users acknowledge that a perfect search experience is not possible, given the size and limitations of the corpus, and we tolerate the imperfections because we are not aware of any better alternative. As two researchers put it, 'College students AND professors might not know that library databases exist, but they sure know Google'.<sup>20</sup>

'There are many examples of AI tools in use without any mention that AI is being used'

## Can we combine the brain with technology?

Machines cannot think, but humans can. One way to assess AI tools is to determine what they are or are not good at. Some human activities lend themselves to automation more than others. Sorting a list into alphabetical or numerical order, for example, is an activity that a spreadsheet can do very easily, but humans can only do slowly and at a high error rate – partly because humans have a limited attention span and find it irritating to sort more than a few records. Does that mean the human brain is inadequate? Hardly, but it does imply that

- 5 human brains do not represent the ideal that all AI research is aimed at emulating. Similarly, humans have very poor information retention skills – we think ourselves clever if we can remember ten phone numbers. Miller's Law,<sup>21</sup> created by a Harvard psychologist, suggests that the number of objects a typical human can hold in memory is just seven.

## What can we meaningfully ask of AI?

Questions we ask of AI tools may have different criteria to scientific research questions. The corpus-based approach using a training set, as described above, uses the process of inductive reasoning. This is the kind of thinking that states 'the sun rose yesterday, the sun rose today, so the chances are the sun will rise tomorrow'. Now, philosophers will argue that inductive reasoning is not scientific. Just because the sun rose yesterday does not mean the sun will rise tomorrow. We would like some external proof to enable us to sleep more peacefully. Inductive reasoning is well described by Eric Larson.<sup>22</sup>

However, for the purpose of AI tools as described here, inductive reasoning may be adequate, indeed ideal. The goal is to use existing evidence to predict a likely inference. Typically, we look to provide good quality results that are better than a human could achieve without the tool. 'Better' here meaning at least as good quality as a human but delivered faster, or better quality with no loss of time compared to a manual process, or both. Hence, for the purpose of AI in this context, you can ask an algorithm if the sun will rise tomorrow, and the machine will give you a workable answer for practical purposes.

To state that narrow AI tools make use of inductive reasoning may seem obvious, yet it is frequently ignored when humans assess the results of a machine-based process. For self-driving cars, an error rate of one in a thousand might mean abandoning the whole project. For spam checking and spell-checking, a much higher error rate may be good enough to use the tool.

## Are we AI literate?

Long and Magerko<sup>23</sup> define AI literacy as 'a set of competencies that enable individuals to critically evaluate AI technologies, communicate and collaborate effectively with AI, and use AI as a tool'. Here is an essential role for the information professional. Millions of people use Google every day, but there is a difference between unthinking use and critical awareness. They further define over 30 relevant factors, of which just the first five are skills I believe to be essential to the assessment and recommendation of AI tools:

1. The ability to distinguish between tools that use or do not use AI.
2. Analyse differences between human and machine intelligence.
3. Identify various technologies that use AI.
4. Distinguish between general and narrow AI.
5. Identify problem types that AI excels at and problems that are more challenging for AI.

To be specific, the skills outlined here do not, I believe, require the ability to code. Given the increasing use of AI tools, it is becoming more difficult to distinguish tools based on human or machine judgement (skill 1). Perhaps this skill will eventually be replaced by skill 5, the ability to identify problem types that lend themselves to an AI-based solution.

'the skills outlined here do not ... require the ability to code'

## AI use in academic contexts

This section looks at some areas where AI tools are currently in use in the scholarly workflow.

## Spell-check

The spell-check tool provided with many common word processors is an example of a widely used and generally accepted algorithm, or collection of algorithms. Users acknowledge (and frequently complain) that spell checkers do not detect all errors that would be detected by a human.

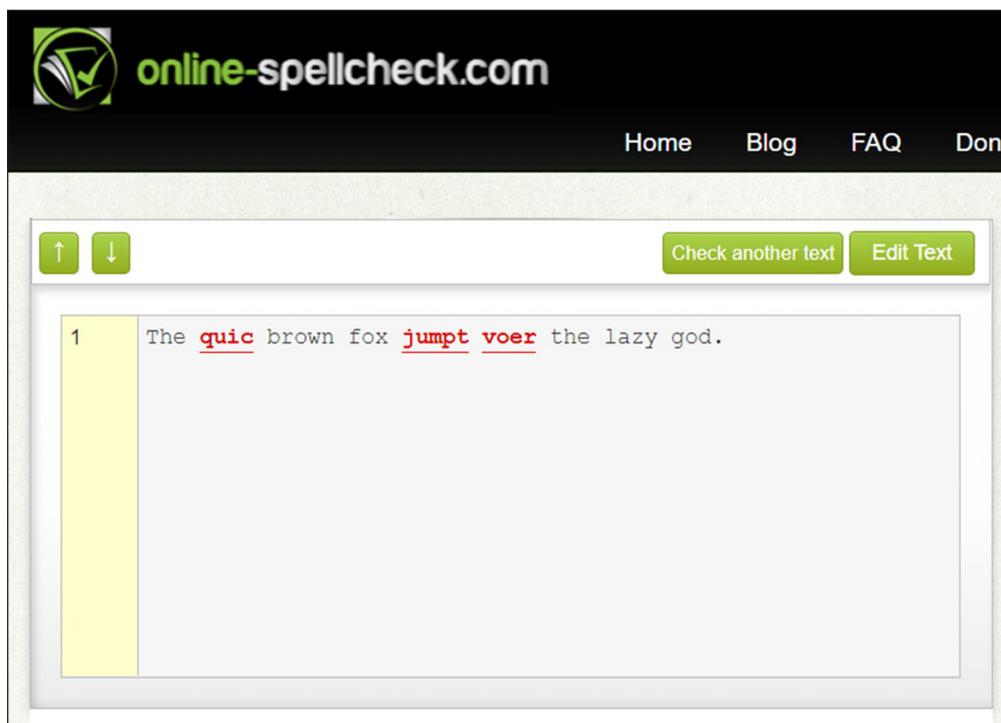


Figure 2. A typical spell checker displaying the limitations of a context-free tool<sup>24</sup>

Users of spell checkers have learned to live with their biggest drawback: that most spell checkers accept a word that corresponds with a term in the dictionary, even if it is the wrong word in context. As shown in Figure 2, the spell checker had no difficulty finding a misspelling for 'quick', but any English speaker would know that the last word should be 'dog', not 'god'. Nonetheless, the use of a spell checker can comprehensively and consistently identify transposed letters in words. The limitations are known and tolerated.

## Spam check

Checking for spam e-mails is one of the most widespread uses of AI. Spam checks use a mixture of word- and phrase-checking to identify a likely spam message. A variety of checks are run, including:

- Is this an unfamiliar sender ID?
- Does the e-mail include terms such as 'offer' or 'bargain'?

As with spell-checking tools, spam checkers are imperfect but widely accepted, because the alternative, of repeatedly reviewing and deleting irrelevant emails, would make the use of e-mail difficult if not impossible. Users tolerate the small number of false positives (e-mails wrongly identified as spam). Jenna Burrell<sup>25</sup> differentiates various kinds of opacity in algorithms and reveals some interesting details about the criteria used to detect spam, but does not mention the corpus dimension of spam checking: e-mails from an e-mail address not in the individual's set of e-mail contacts is more likely to be spam.

## Plagiarism detection

Plagiarism detection, such as Turnitin, Copyleaks and others, can use string or semantic matching, or both. The most common form is simply checking for string matches. A simple plagiarism check can be done against published articles by simply searching for a string, such as a full article title, in Google – the system typically finds a match (if one exists) in less than a second, see Figure 3.

The screenshot shows a Google search results page. The search query is "intravenous ibuprofen for the treatment of post-operative pain". The results include a snippet from a PubMed article: "Furthermore IV-Ibuprofen was safe and well tolerate. Consequently we consider appropriate that protocols for management of postoperative pain include **IV-Ibuprofen 800 mg every 6 hours** as an option to offer patients an analgesic benefit while reducing the potentially risks associated with morphine consumption." Below the snippet is a link to the full article: "Intravenous Ibuprofen for Treatment of Post-Operative Pain".

Figure 3. Google search for an article title

In recent years, plagiarism checks have become more sophisticated. While most common search engines can find strings of characters very efficiently, it is more challenging to find semantic matches. If the plagiarist uses the same ideas but changes a few of the words for common synonyms, the plagiarism is (currently) far less likely to be detected. This limitation does not prevent plagiarism tools being widely used by many academic publishers.

## Discovery

One of the longest-established uses of AI is in content discovery. This can range from the simple recommender tool ('if you like X, you will like Y') to much more sophisticated recommenders that identify concepts in an article and match those concepts to other articles. Figure 4 shows an example from the Cambridge University Press content collection, linking book chapters to other book chapters and to articles:<sup>26</sup>

The screenshot shows a Cambridge Core search results page for the chapter "24 - Machine learning" from the book "Python Programming for Biology". The chapter is published online by Cambridge University Press on 05 February 2015. The page includes a summary of the chapter content and links to related content, such as "Non-probabilistic Classifiers" and "Support vector machines".

Figure 4. Example of a recommender system in action

Elaborations of the discovery tool include an alerting service, which finds new articles, essentially by replicating the search with a date filter, to look only for articles published on a subject in the last six months or six weeks on a topic. Recommender tools are common on most academic discovery platforms.

## Impact

Citations are one way of assessing the relative worth of an article – if it has been widely cited, it must be significant. Citation indexes for academic journals were introduced in 1955 by Eugene Garfield.<sup>27</sup> But, of course, citations are contentious as indicators of quality. An article might be cited because the writer thinks the source article is incorrect. Citations for certain types of articles, such as review articles, are always higher than for research articles.

One reason why citations became widely adopted as a metric for article quality is that they can be counted. A human judgement ('is this article significant?') is thereby represented, however imperfectly, by an arithmetic tool. However, it is now recognized that a simple count of citations is unsatisfactory, and several modifications of the tool have been proposed, for example, the Hirsch or H-index.<sup>28</sup>

AI tools have enabled a more sophisticated analysis of citations. Tools are available, for example from scite.ai,<sup>29</sup> Scholarcy<sup>30</sup> (see Figure 5) and Semantic Scholar,<sup>31</sup> that not only identify citations, but show if the citation supports or refutes a statement.<sup>32</sup>

'AI tools have enabled a more sophisticated analysis of citations'

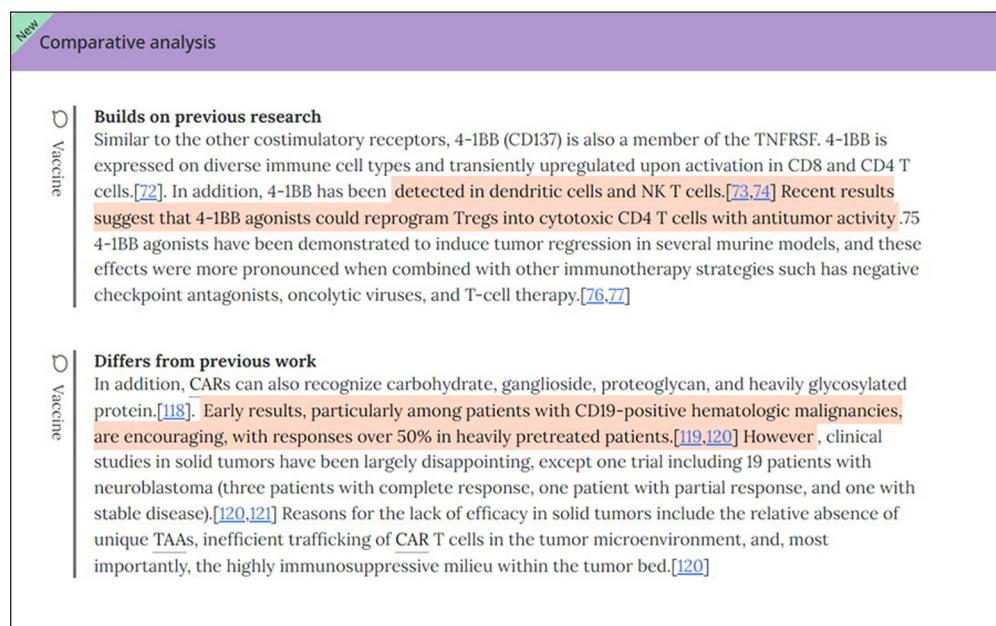


Figure 5. Categorizing citations into supporting or differing from earlier research<sup>33</sup>

Currently, few researchers will be aware of the wealth of tools available to them in this area.

## The need for analytics

Any intervention in the academic workflow can only be assessed for robustness, efficacy and accuracy, if its impact is evaluated. This is as true for AI tools as for any other attempt to improve the process. Accordingly, both libraries and publishers have a responsibility to identify if, and how, AI tools are used, with an attempt to identify the impact of those interventions. However, many libraries do not carry out such studies. According to a 2021 survey of library analytics practice<sup>34</sup> the greatest barriers to data analysis (interpreted broadly to include bibliometrics, studies of user behaviour and such like) by libraries were:

- 61% lack of time
- 54% lack of expertise
- 52% lack of personnel

Unfortunately, all these justifications for inaction are ultimately self-defeating. If a poor-quality AI utility is adopted by the library, it will take more, rather than less time to manage its use and quite possibly lead to misunderstanding and corresponding negative feedback. Introducing a tool without capturing the data to assess its success cannot be a sensible procedure.

## Automating metadata

Machines have difficulty with ambiguity, while humans are tolerant of inconsistencies and small errors. However, increased use of AI has partially resolved this distinction. Today, keying 'shakespeare' into Google results in the tool automatically suggesting the closest match from its index, see Figure 6.

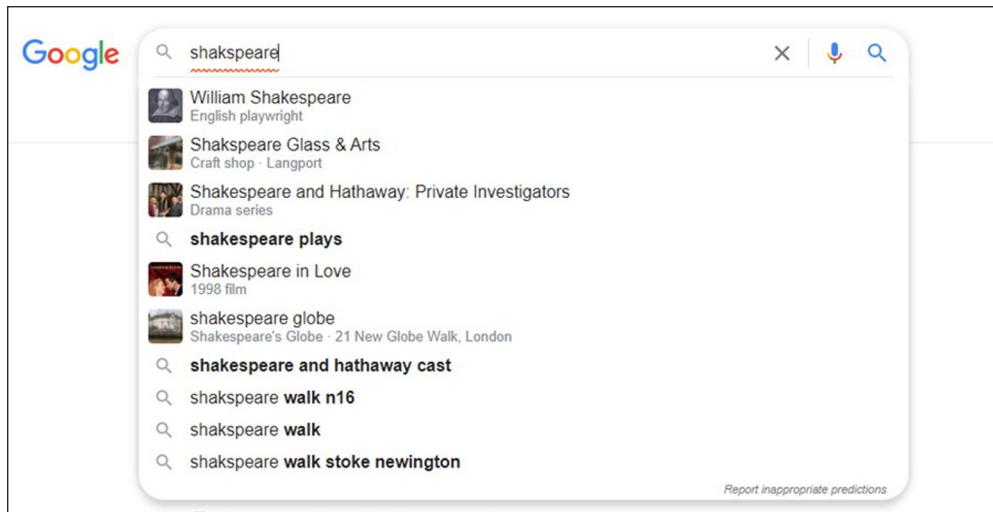


Figure 6. Google search 4 May 2022

This is rather similar to the way we tolerate a high level of incompleteness and errors in spoken dialogue – we guess what a speaker is trying to say. Similarly, the 'search ahead' feature in Google, see Figure 7, and other search engines attempts to guess what the user intends:

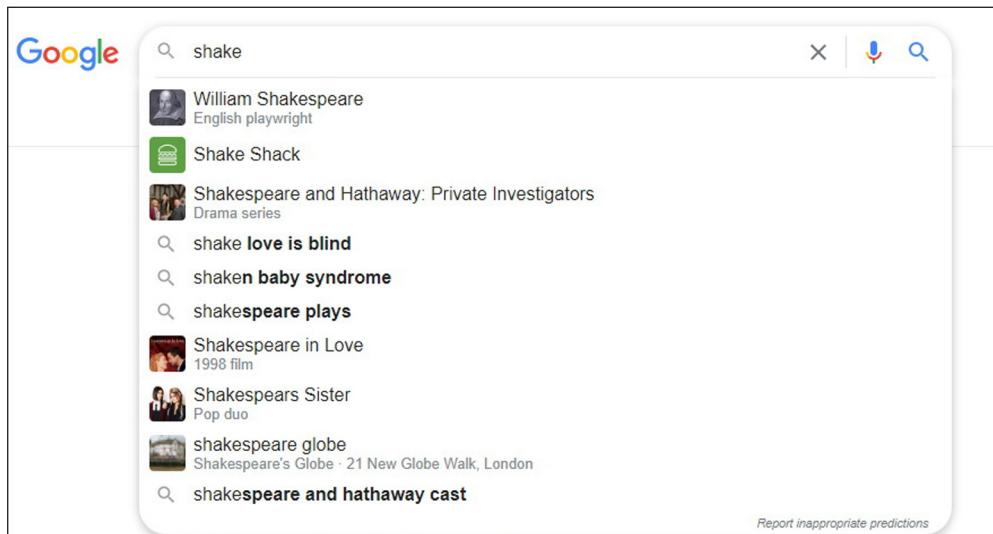


Figure 7. Google predictive search 4 May 2022

The system may be correct, or it may on occasion be wrong, but we tolerate its errors because most of the time it works and saves us effort. Humans have learned to live with the imperfect world of AI.

'Humans have learned to live with the imperfect world of AI'

## Potential misuses of AI

Once developed, AI tools can be extended to domains where their validity is greatly reduced. There are several examples of this overextension of AI tools, with predictably irrelevant or meaningless results. A ranking website, Academic Influence, provides metrics for ranking departments, scholars and faculties with the slogan 'better rankings for a better education'. The site also contains a list of 'the most influential people

for the years 4000 B.C. to 2022', with Aristotle in first place, ahead of Plato, Marx and Kant. Shakespeare is listed in sixth place, with his plays and sonnets listed under his 'academic contributions'.<sup>35</sup> In this case, a tool developed for the comparison of current scholarship has been overextended back by several hundred years to a time before scholarly communications existed, see Figure 8 – and yet the site claims that their metrics are built using 'sanity checks': 'we make sure the rankings make sense by performing "sanity checks" against other independent information sources such as periodicals, journals, and global media outlets'.

The screenshot shows a web page from 'ACADEMIC INFLUENCE'. At the top, there is a navigation bar with links: School Rankings, People Rankings, Build Your Own, Subjects, Resources, Magazine, Mission, Sign In, and a search icon. Below the navigation bar, there is a section titled 'Other Resources About William Shakespeare' which includes a link to 'en.wikipedia.org'. The main content area is titled 'What Are William Shakespeare's Academic Contributions?'. It contains a brief introduction about Shakespeare's academic work in literature and fields like Sonnet 48, The Taming of the Shrew, Sonnet 32, and Sonnet 37.

Figure 8. Entry for William Shakespeare from 'the 50 most influential scholars of all time'

Trying to establish the greatest thinkers in world history via an algorithm for researchers and academics is unlikely to produce trustworthy results.

## The need for a sanity check

There is undoubtedly a need for sanity checking of AI-based tools, and humans are necessary to carry this out. Algorithms in narrow AI have no knowledge of the real world. A system that can differentiate images of cats from images of dogs could still not define what a cat or a dog is, nor could such a system identify any other animal species. However, it is tempting to apply an algorithm to a corpus way beyond a viable scope. However clever the algorithm, there is a need for a human to check that the results correspond with a common-sense view. What is meant by common sense here is quite specific, and quite limited. For example, consider an algorithm that applies subject tags to an academic article. This algorithm provides a probability ranking for the subjects physics, chemistry and politics. Using the algorithm, the following results were obtained, using a ranking between 0 and 1, see Table 1.

'A system that can differentiate images of cats from images of dogs could still not define what a cat or a dog is'

Content	physics	chemistry	politics
Article 1	0.65	0.54	0.12
Article 2	0.45	0.73	0.19

Table 1. Typical predictive scores for subject tagging

Clearly, article 1 looks to be more about physics than anything else, while article 2 seems to be obviously about chemistry. However, the algorithm has found traces of politics content in both articles. The determination of a suitable threshold, below which the user should state 'this article is not about politics', needs to be determined by common sense – or by using a subject-matter expert to identify what the threshold should be. In this case, the machine delivers a result, but in all probability the result for politics can be discounted.

11 One of the most widely used metrics for text-based AI is the F1 score,<sup>36</sup> which measures the mean of recall and precision, usually shown on a scale between 0 (no accuracy) and 1 (perfect accuracy). However, the F1 score has limitations which are easy to recognize using common sense. Harikrishnan<sup>37</sup> gives an example of a pregnancy test of 100 women, which identifies five as pregnant when they are not (false positive), and ten as not pregnant when they are pregnant (false negative). A machine-based algorithm that resulted in these figures would have an F1 score of 0.8, which in other contexts might represent an acceptable score, but would certainly not be adequate for a pregnancy test.

This is where the information professional has a key role. Some idea of context, of what is or is not required in the situation, is vital for ensuring that any tool delivers relevant results. Are the results relevant in context?

An example of the F1 score in use for subject tagging is described by Goh.<sup>38</sup> This study compares humans with a machine used to classify articles by subject. The machine outperforms the postgraduates by delivering a significantly more accurate set of tags. Even more impressive, it took one postgraduate two hours to classify 247 abstracts, compared to five seconds for a machine to complete the same exercise.

A comparison of the subject tagging between the machine and the humans shows immediately that the machine consistently delivers better results than the human taggers, but the most significant inference of the study is implied rather than explicitly stated. For the purpose of subject tagging, humans were found in this controlled study to have an average F1 score in the region of 0.5 or less, while the machine result was considered to be usable with an F1 score of around 0.7. While this figure at first glance seems poor (given that a perfect score would be 1), the implication is that the F1 score should be interpreted in context, not as an absolute measure. In other words, when comparing machine-based with human results, we should be considering relative, rather than absolute, measurements. If a machine delivers a better result than what is achievable by hand, it makes sense to adopt the machine solution immediately.

**'If a machine delivers a better result ... it makes sense to adopt the machine solution immediately'**

Some sanity checks can be built into the tool itself. For example, a tool to identify peer reviewers could helpfully provide an indication when an article is submitted that is outside the corpus used to identify reviewers. A tool to identify the most relevant journal for an article could have a result of 'no suitable match found' if an article is submitted to the tool that is outside the subject areas of the corpus.

## The corpus and bias

Another key role that information professionals can play in the evaluation of AI tools is the awareness of potential and actual bias. Any corpus contains bias. Bias is typically independent of any AI tools. All real-world data is inevitably biased. Even seemingly neutral collections reveal unconscious bias. For example, it might be assumed that PubMed, a collection of millions of scholarly articles published on biomedical topics over the last 50 years, would comprise a statistically valid corpus, yet there are more male than female authors in PubMed. Is this surprising? A study of gender disparity in medical articles<sup>39</sup> found this was the case over the last 20 years. Another article<sup>40</sup> shows a revealing graph of male and female authorship of articles in science journals since 1955. While the proportion of female authorship is growing, males continue to author the majority of science articles.

Of course, once bias is recognized, it is possible to take steps to work around this bias, but lack of awareness of bias means that there is an important role for information professionals when recommending these tools.

## Corpus size

It is difficult to give absolute figures, but statistically based and reliable AI tools require at least several hundred documents in the training set, and a minimum of several thousand documents in the corpus. Depending on the goal, the training set may need to be larger if the question asked is less straightforward than differentiating between numbers or letters.

## The algorithm

Any algorithm should be open in the sense that its basic methodology is clearly stated. This should not require any breach of commercial confidence for paid software and has the benefit that potential bias can therefore be revealed. The alternative, without any explanation of the methodology, is the dreaded 'black box', a tool that must be trusted rather than understood. An example of a methodology statement is 'we find peer reviewers by identifying similar articles to your submission. Then we identify the authors of those articles.'. This is how most AI tools to identify peer reviewers work, for example the Web of Science Reviewer Locator.<sup>41</sup> What constitutes a clearly defined algorithm is well explained on the 50 Examples page 'Background: Algorithms'.<sup>42</sup>

## Manual checking

It is useful, but by no means a complete assessment, to try out the tool directly. By inputting an example or two with a known prediction, some idea can be obtained of the capabilities. It is surprising how frequently this technique can reveal assumptions on the part of the software developers that were not made clear when the tool was delivered.

## Evaluation and metrics

Asking a couple of colleagues to have a look at the product is not a full evaluation. If using human criteria to evaluate the tool, give thought to suitable criteria for comparing human and machine performance – see, for example, Ewerth.<sup>43</sup>

When building or adjusting websites, A/B tests are often used. An A/B test is a randomized trial widely used in website development, in which two versions of a variable, such as a web page layout, are shown to different groups of users, and their resulting behaviour is measured. Neither group is aware of the trial or of the other version. In this way, like a randomized control trial, it is possible to identify which version has the greatest impact. A/B tests on the live site are the best way to evaluate the impact of any change.<sup>44</sup> This is because:

- evaluations are made with data rather than by guesswork
- the test gives the response of real users
- the results enable the estimation of metrics of success – what is an acceptable goal (rather than an absolute goal).

Similarly, for many AI tools, a methodology based on the A/B test is feasible, if complex, to provide a solid assessment. For example, a comparison of machine-based results and human-generated results could be carried out.

As for human evaluation, all humans are not equal at this task. Consider who are the best people to evaluate this tool. Should it be the person managing the process, or should it be the end user? It is a well-established principle in website design that the best way to evaluate software is to test it with real users, not with the software team who built the tool, or with the people tasked with managing the delivery of the tool.

When humans are used to evaluate a tool, there is the question of how many checks are required. If we ask the machine to use a training set of, say, 500 documents to determine the parameters for the exercise, does it make sense to judge the results by a human looking at two or three examples? What is the level of human agreement?

## Credibility

What role does the information professional play in all this? Most of the criteria described above could be checked directly by the user, that is, the researcher, but most researchers have neither the time nor the knowledge to make measured comparisons of different tools. Without a solid analytical framework, humans tend to rely on instinct, which could be described as an internal assessment mechanism – they instinctively trust (or do not trust) a familiar methodology, or tools they have used before. The role for the information professional in all this is providing credibility: providing users with external validations that enable them to trust a tool and to deploy it with confidence. Researchers will, for the most part, look for an external validation of a tool that they can trust. The information professionals provide the credibility, based on their detailed evaluation.

'The role for the information professional in all this is providing credibility'

## The AI toolkit: a framework for evaluation of AI tools

Here, in summary, is a toolkit for information professionals appraising any AI tool. Although making use of Long and Magerko's idea of AI literacy, the requirements here are much more specific.

### **Goal**

1. What is a realistic goal? Expecting perfection for an AI utility is impossible. AI tools based on a training set cannot have 100% accuracy. Nonetheless, the accuracy they provide should be considerably greater than using humans for the same task.

### **Corpus**

2. Is the corpus large enough? Is the training set large enough?
3. What are the start and end dates for the data in the corpus? Does this matter?
4. Who chose the corpus, when was it chosen and for what purpose? Details of the corpus used, like the data for a research article, should be publicly stated and accessible.
5. What is the corpus bias?
6. Is the tool likely to raise diversity, equality and/or inclusion issues?
7. Is personal data captured and reused?

### **Algorithm**

8. Have the developers provided a single-sentence summary of the methodology behind the algorithm?

### **Evaluation and Metrics**

9. Have I measured the current process before introducing any change, for example, time taken, number of errors?
10. Who to evaluate: end users or subject-matter experts, or both? Internal or external?
11. What metrics will be used to evaluate the tool? The F1 score, if used, must be interpreted in context.

### **Sanity check**

12. Sanity check/common sense: Have the developers built in 'common-sense' limitations to prevent the algorithm being applied too widely? Am I asking a meaningful question? Is this a feasible exercise?
13. Does the tool provide feedback when a question is out of scope?
14. Based on the checks above, is the tool fit for purpose?

## Dissemination

Is there easy-to-read documentation and guidance for new users that explains in simple terms how to use the tool and how it improves on current processes?

## Feedback

Does the tool provide a feedback loop so it can be improved over time?

## Conclusion

To make the best use of AI tools in publishing requires not only high-quality software, but also a critical awareness of the context in which the tool will be used. By following a template, and asking the right questions, those responsible for recommending and assisting in the take-up of AI tools can ensure a much higher success rate for this technology. All of us, often without realizing it, have developed in our everyday life an unconscious heuristic approach to working with AI and with those tools that make use of AI around us. If we ignore AI, we miss a host of benefits that can make us work more effectively. If we make use of AI tools uncritically, we risk discrediting a whole area of new technology. By using this framework, information professionals, without being developers, can become qualified to assess AI tools with confidence.

'By using this framework, information professionals ... can become qualified to assess AI tools with confidence'

### Abbreviations and Acronyms

A list of the abbreviations and acronyms used in this and other *Insights* articles can be accessed here – click on the URL below and then select the 'full list of industry A&As' link: <http://www.uksg.org/publications#aa>.

### Competing interests

The author has formerly provided services for UNSILO, a company mentioned in this article.

### References

1. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 3rd edition (Pearson, 2016).
2. Ray Kurzweil, "A Wager on the Turing Test: Why I Think I Will Win," *Kurzweil, Kurzweilai.net*, April 9, 2002, <https://www.kurzweilai.net/a-wager-on-the-turing-test-why-i-think-i-will-win> (accessed 23 September 2022).
3. Rory Cellan-Jones, "Stephen Hawking Warns Artificial Intelligence Could End Mankind," *BBC News*, December 2, 2014, sec Technology, <https://www.bbc.com/news/technology-30290540> (accessed 23 September 2022).
4. A. M. Turing, "Intelligent Machinery, A Heretical Theory\*," *Philosophia Mathematica* 4, no. 3 (September 1, 1996): 256–60, DOI: <https://doi.org/10.1093/philmat/4.3.256> (accessed 23 September 2022).
5. Nicholas Carr, *The Shallows: How the Internet Is Changing the Way We Think, Read and Remember*, Main-Re-issue edition (London: Atlantic Books, 2020).
6. Elon Musk, "Elon Musk: 'With Artificial Intelligence We Are Summoning the Demon,'" *Washington Post*, October 24, 2014, <https://www.washingtonpost.com/news/innovations/wp/2014/10/24/elon-musk-with-artificial-intelligence-we-are-summoning-the-demon/> (accessed 23 September 2022).
7. Kate Crawford, *Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence* (New Haven: Yale University Press, 2021). DOI: <https://doi.org/10.12987/9780300252392>
8. Cathy O'Neil, *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*, 01 edition (Penguin, 2016).
9. Erik J. Larson, *The Myth of Artificial Intelligence: Why Computers Can't Think the Way We Do* (Cambridge, Massachusetts: Belknap Press, 2021). DOI: <https://doi.org/10.4159/9780674259935>
10. Yann LeCun, Corinna Cortes, and Christopher J C Burges, "The MNIST Database," THE MNIST DATABASE, <http://yann.lecun.com/exdb/mnist/> (accessed 23 September 2022).
11. Natalia Domagala and Hannah Spiro, "Engaging with the Public about Algorithmic Transparency in the Public Sector," *Centre for Data Ethics and Innovation Blog*, June 21, 2021, <https://cdei.blog.gov.uk/2021/06/21/engaging-with-the-public-about-algorithmic-transparency-in-the-public-sector/> (accessed 23 September 2022).
12. Domagala and Spiro, "Engaging with the Public."
13. European Parliament. Directorate General for Parliamentary Research Services. *A Governance Framework for Algorithmic Accountability and Transparency* (LU: Publications Office, 2019), <https://data.europa.eu/doi/10.2861/59990> (accessed 23 September 2022).
14. Rob Merrick, "Fears of Another A-Level-Style Fiasco as Scrutiny of Policies Made by Computer Are Ditched Following Brexit," *The Independent*, February 10, 2022.
15. Pedro Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*, 1st edition (London: Penguin, 2017).

16. "Artificial General Intelligence," in *Wikipedia* (Retrieved 9 May 2022),  
[https://en.wikipedia.org/w/index.php?title=Artificial\\_general\\_intelligence&oldid=1086964857](https://en.wikipedia.org/w/index.php?title=Artificial_general_intelligence&oldid=1086964857) (accessed 23 September 2022).
17. "What Is Supervised Learning?," IBM Cloud Learn Hub, August 19, 2020,  
<https://www.ibm.com/cloud/learn/supervised-learning> (accessed 23 September 2022).
18. Pandu Nayak, "Understanding Searches Better than Ever Before," Google (blog), October 25, 2019,  
<https://blog.google/products/search/search-language-understanding-bert/> (accessed 23 September 2022).
19. Motahhare Eslami et al., "'I Always Assumed That I Wasn't Really That Close to [Her]': Reasoning about Invisible Algorithms in News Feeds," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15* (New York, NY, USA: Association for Computing Machinery, 2015), 153–62, DOI:  
<https://doi.org/10.1145/2702123.2702556> (accessed 23 September 2022).
20. Mary Shultz, "Comparing Test Searches in PubMed and Google Scholar," *Journal of the Medical Library Association: JMLA* 95, no. 4 (October 2007): 442–45, DOI:  
<https://doi.org/10.3163/1536-5050.95.4.442> (accessed 23 September 2022).
21. G. A. Miller, "The Magical Number Seven plus or Minus Two: Some Limits on Our Capacity for Processing Information," *Psychological Review* 63, no. 2 (March 1956): 81–97,  
<https://pubmed.ncbi.nlm.nih.gov/13310704/> DOI:  
<https://doi.org/10.1037/h0043158> (accessed 27 September 2022).
22. Larson, *The Myth of Artificial Intelligence*.
23. Duri Long and Brian Magerko, "What Is AI Literacy? Competencies and Design Considerations," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu HI USA: ACM, 2020)*, 1–16, DOI:  
<https://doi.org/10.1145/3313831.3376727> (accessed 23 September 2022).
24. "[Online-Spellcheck.Com](https://www.online-spellcheck.com/),"  
<https://www.online-spellcheck.com/> (accessed 23 September 2022).
25. Jenna Burrell, "How the Machine 'Thinks': Understanding Opacity in Machine Learning Algorithms," *Big Data & Society* 3, no. 1 (January 6, 2016): 2053951715622512, DOI:  
<https://doi.org/10.1177/2053951715622512> (accessed 23 September 2022).
26. "Cambridge Core, Recommender Links for Book Chapter," *Cambridge Core*,  
<https://www.cambridge.org/core/books/abs/python-programming-for-biology/machine-learning/570B575C26034A8CB9A7AF7E17A795AB/> (accessed 23 September 2022).
27. Eugene Garfield, "Citation Indexes for Science," *Science* 122, no. 3159 (July 15, 1955): 108–11, DOI:  
<https://doi.org/10.1126/science.122.3159.108> (accessed 23 September 2022).
28. J. E. Hirsch, 'An Index to Quantify an Individual's Scientific Research Output', *Proceedings of the National Academy of Sciences* 102, no. 46 (November, 15 2005): 16569–72, DOI:  
<https://doi.org/10.1073/pnas.0507655102> (accessed 23 September 2022).
29. "Scite: See How Research Has Been Cited," [scite.ai](https://scite.ai),  
<https://scite.ai/> (accessed 23 September 2022).
30. "Scholarcy," Scholarcy|The long-form article summariser,  
<https://www.scholarcy.com/> (accessed 23 September 2022).
31. "Semantic Scholar|AI-Powered Research Tool,"  
<https://www.semanticscholar.org/> (accessed 23 September 2022).
32. Marco Valenzuela, Vu Ha, and Oren Etzioni, "Identifying Meaningful Citations," in *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, 6,  
<http://aaai-website.s3.amazonaws.com/publications/ValenzuelaHaMeaningfulCitations.pdf> (accessed 27 September 2022).
33. "Scholarcy".
34. "2021 Trends in Library Analytics," EBSCO Information Services, Inc., December 13, 2021,  
<https://www.ebsco.com/blogs/ebscopost/2021-trends-library-analytics> (accessed 23 September 2022).
35. "William Shakespeare," *Academic Influence*,  
<https://academicinfluence.com/people/william-shakespeare-1> (accessed 23 September 2022).
36. "F-Score," *Wikipedia*, May 9, 2022,  
<https://en.wikipedia.org/w/index.php?title=F-score&oldid=1086969326> (accessed 23 September 2022).
37. N.B. Harikrishnan, "Confusion Matrix, Accuracy, Precision, Recall, F1 Score," *Analytics Vidhya* (blog), December 10, 2019,  
<https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd> (accessed 23 September 2022).
38. Yeow Goh et al., "Evaluating Human versus Machine Learning Performance in Classifying Research Abstracts," *Scientometrics* 125 (July 18, 2020): 1197–1212, DOI:  
<https://doi.org/10.1007/s11192-020-03614-2> (accessed 23 September 2022).
39. Karla Bernardi et al., "Gender Disparity in Authorship of Peer-Reviewed Medical Publications," *The American Journal of the Medical Sciences* 360, no. 5 (November 2020): 511–16, DOI:  
<https://doi.org/10.1016/j.amjms.2019.11.005> (accessed 23 September 2022).
40. Tyler Machado, Molly Callahan, and Eunice Esomonu, "Do Women Publish Less than Men in Scientific Fields?," *News @ Northeastern*, March 5, 2020,  
<https://news.northeastern.edu/2020/03/05/do-women-publish-less-than-men-in-scientific-fields-turns-out-scientists-have-been-asking-the-wrong-question/> (accessed 23 September 2022).
41. "Web of Science Reviewer Locator," Clarivate,  
<https://clarivate.com/products/scientific-and-academic-research/research-publishing-solutions/web-of-science-reviewer-locator/> (accessed 23 September 2022).

42. "Background: Algorithms," 50 Examples 1.0 Documentation, <https://fiftyexamples.readthedocs.io/en/latest/algorithms.html> (accessed 23 September 2022).
43. Ralph Ewerth et al., "'Are Machines Better Than Humans in Image Tagging?' – A User Study Adds to the Puzzle," *Advances in Information Retrieval*, ed. Joemon M Jose et al., Lecture Notes in Computer Science (Cham: Springer International Publishing, 2017), 186–98, DOI: [https://doi.org/10.1007/978-3-319-56608-5\\_15](https://doi.org/10.1007/978-3-319-56608-5_15) (accessed 23 September 2022).
44. Ron Kohavi, Diane Tang, and Ya Xu, *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing* (Cambridge: Cambridge University Press, 2020), DOI: <https://doi.org/10.1017/9781108653985> (accessed 23 September 2022).

**Article copyright: © 2022 Michael Upshall. This is an open access article distributed under the terms of the Creative Commons Attribution Licence, which permits unrestricted use and distribution provided the original author and source are credited.**



Corresponding author:

Michael Upshall

Consultant, GB

E-mail: michael@consultmu.co.uk

ORCID ID: 0000-0003-1115-6847

To cite this article:

Upshall M, "An AI toolkit for libraries," *Insights*, 2022, 35: 18, 1–16; DOI: <https://doi.org/10.1629/uksg.592>

Submitted on 08 June 2022

Accepted on 18 July 2022

Published on 01 November 2022

Published by UKSG in association with Ubiquity Press.



# An enhanced approach for sentiment analysis based on meta-ensemble deep learning

Rania Kora<sup>1</sup> · Ammar Mohammed<sup>1</sup>

Received: 31 December 2022 / Revised: 27 January 2023 / Accepted: 17 February 2023  
© The Author(s) 2023

## Abstract

Sentiment analysis, commonly known as “opinion mining,” aims to identify sentiment polarities in opinion texts. Recent years have seen a significant increase in the acceptance of sentiment analysis by academics, businesses, governments, and several other organizations. Numerous deep-learning efforts have been developed to effectively handle more challenging sentiment analysis problems. However, the main difficulty with deep learning approaches is that they require a lot of experience and hard work to tune the optimal hyperparameters, making it a tedious and time-consuming task. Several recent research efforts have attempted to solve this difficulty by combining the power of ensemble learning and deep learning. Many of these efforts have concentrated on simple ensemble techniques, which have some drawbacks. Therefore, this paper makes the following contributions: First, we propose a meta-ensemble deep learning approach to improve the performance of sentiment analysis. In this approach, we train and fuse baseline deep learning models using three levels of meta-learners. Second, we propose the benchmark dataset “Arabic-Egyptian Corpus 2” as an extension of a previous corpus. The corpus size has been increased by 10,000 annotated tweets written in colloquial Arabic on various topics. Third, we conduct several experiments on six benchmark datasets of sentiment analysis in different languages and dialects to evaluate the performance of the proposed meta-ensemble deep learning approach. The experimental results reveal that the meta-ensemble approach effectively outperforms the baseline deep learning models. Also, the experiments reveal that meta-learning improves performance further when the probability class distributions are used to train the meta-learners.

**Keywords** Ensemble learning · Ensemble deep learning · Ensemble methods · Deep learning · Sentiment analysis

## 1 Introduction

The power of social media for expressing opinions about events, topics, people, services, or products has expanded due to the growth of user-generated content on platforms (Naresh and Venkata Krishna 2021). Hence, analyzing this huge amount of social media data can help better understand public opinions and trends and effectively make important decisions by classifying the opinions and feelings expressed in the text and determining their polarity as positive, negative, or neutral (Mejova 2009).

In the literature, several research efforts have been introduced to approach sentiment analysis using machine learning (Pontiki et al. 2016; Ahmed et al. 2013; Duwairi et al. 2014; Shoukry and Rafea 2012; Alomari et al. 2017). Extended efforts have used deep learning to handle bigger data and improve the classification’s performance against classical machine learning models (Mohammed and Kora 2019; Chen et al. 2018; Pontiki et al. 2016; Heikal et al. 2018; Baly et al. 2017; Rojas-Barahona 2016). Deep learning techniques aim to overcome the limitations and problems of classical learning through efficient approaches in dealing with complex problems, large amounts of data, and its capacity to automatically extract the feature from the text (Habimana et al. 2020; Chan et al. 2020). There are several architectures and models for deep learning approaches when applied to sentiment analysis, such as recurrent neural networks (RNN) (Mitra and Mandal 2019), gated recurrent unit (GRU) (Le et al. 2019), Long Short-Term Memory (LSTM) (Graves

✉ Ammar Mohammed  
ammar@cu.edu.eg

Rania Kora  
rania.kora@pg.cu.edu.eg

<sup>1</sup> Department of Computer Science, Faculty of Graduate Studies for Statistical Researches, Cairo University, Cairo, Egypt

2012), Convolutional Neural Networks (CNN) (Collobert and Weston 2008). However, the main difficulty with deep learning techniques is identifying the most appropriate architectures and models. Usually, deep models require much effort due to tuning the optimal hyperparameters in the search space of the possible hyperparameters, which is a tedious task (Yadav and Vishwakarma 2020). These problems can be overcome by approaching ensemble learning to deep learning. Traditional ensemble learning refers to merging several basic models to build one powerful model (Kumar et al. 2021). Ensemble learning has been successfully applied in many fields, such as image classification (Wang et al. 2013), medical image (Cho and Won 2003; Shipp and Kuncheva 2002), music recognition (Stamatatos and Widmer 2002), malware detection (Shahzad and Lavesson 2013) and text classification (Kulkarni et al. 2018). In the literature, there are several ensemble approaches, like, averaging, boosting, bagging, random forest, and stacking (Zhang and Ma 2012). In deep learning, most ensemble learning is a simple averaging of model (Tan et al. 2022; Mohammadi and Shaverizade 2021; Araque et al. 2017) due to its simplicity and high results. However, the voting-based ensemble method is not a smart method to combine the models because it is biased toward weak models, which can reduce the performance in a lot of problems (Tasci et al. 2021).

To this end, the primary objectives of this research are four-fold. First, we propose a meta-ensemble deep learning approach to boost the performance of sentiment analysis. The proposed approach combines the predictions of several groups of deep models using three levels of meta-learners. In the proposed approach, we achieve diversity in the ensemble by using differences in the training data, the diversity of trained baseline deep learners, and the variation within the fusion of baseline deep models. Second, we propose the benchmark dataset “Arabic-Egyptian corpus”, which consists of 50,000 tweets written in colloquial Arabic on various topics. This corpus is an extended version of the corpus “Arabic-Egyptian corpus” (Mohammed and Kora 2019). Third, we conduct a wide range of experiments on six public benchmark datasets to study the performance of the proposed meta-ensemble deep learning approach on sentiment classification in different languages and dialects. For each benchmark dataset, groups of different deep baseline models are trained on partitions of the trained data. Their best performance is compared with the proposed meta-ensemble deep learning approach. Finally, we show the impact of meta-predictions of the proposed meta-ensemble deep learning approach through different models’ predictions, namely the class label probability distribution and the class label predictions. The main contributions of the paper can be summarized as follows:

- We propose a meta-ensemble deep learning approach to improve the sentiment classification performance that combines three levels of meta-learners.
- We extended the Arabic-Egyptian corpus (Mohammed and Kora 2019) by increasing it to 50k annotated tweets.
- We train several baseline deep models using six public benchmark sentiment analysis datasets in different languages and dialects.
- We conduct a wide range of experiments to study the effect of the meta-ensemble deep learning approach against single deep learning models.
- We compare the effect of the generated predictions of meta-learners involved in the proposed approach to improve the performance.

The paper is structured as follows: Sect. 2 provides a brief overview of the challenges of sentiment analysis and various ensemble learning methods as well as highlighting some of the literature used for ensemble learning in sentiment analysis. Section 3 describes the meta-ensemble deep learning approach. Section 4 shows the experimental results, the evaluation of the baseline deep learning models, and the meta-ensemble deep learning approach in each of the different benchmark datasets. Finally, Sect. 5 concludes the paper and suggests future research directions.

## 2 Related work

Through sentiment analysis, we can obtain important information that helps in making decisions, solving problems, managing crises, correcting misconceptions, providing desired products and services, interacting with consumers on their terms, improving product and service quality, discovering new marketing strategies and increasing sales (Tuysuzoglu et al. 2018). Despite its benefits, sentiment analysis is an extremely difficult task due to several challenges and problems (Cambria et al. 2017). First, the problem of identifying the subjective parts of the text: The same word can be treated as subjective in one context, while it might be objective in some other. This makes it challenging to distinguish between subjective and objective (sentiment-free) texts. For instance: “The writer’s language was very crude,” and “Crude oil is extracted from the sea-beds”. Second, the problem of domain Dependence: In other contexts, the same sentence can indicate something quite different. The word unpredictable is negative in the domain of movies, but when used in another context, it has a positive connotation. For instance: “The movie was too slow and too long”, “I love long pasta”. Third, the problem of sarcasm Detection: Sarcasm sentences use positive words to convey a negative opinion about a target. For instance: “Nice perfume. You must be marinated in it”. Fourth, the problem of thwarted

Expressions: In some sentences, the polarity of the text is determined by a small portion of the text. For instance: “Although I’m tired, the day is great.” Fifth, the problem of indirect Negation of Sentiment: Such negations are not easily defined because they do not contain “no,” “not,” etc. Sixth, the problem of order Dependence: When the words are not considered independent. For instance, “A is better than B”. Seventh, the problem of entity Recognition: A text may not always refer to the same entity. For instance, “I hate Samsung, but I like OPPO”. Eighth, the problem of identifying Opinion Holders: All written in a text is not always the author’s opinion. For instance, when the author quotes someone. Ninth and finally, the problem of associating sentiment with specific keywords: Many statements express very strong opinions, but it is impossible to identify the source of these sentiments. Generally, sentiment analysis can occur at three levels: Sentence, Document, and Aspect/Feature. At the sentence level, the task of this level is sentence by sentence and decides whether each sentence represents a neutral, positive, or negative opinion. At the document level, this analysis level identifies a document’s overall sentiment and categorizes it as negative or positive. At the aspect level (also known as a word or feature level), this level of analysis aims to discover sentiments on entities and/or their aspects (Wagh and Punde 2018).

In recent years, ensemble learning has been considered one of the most successful techniques in machine learning (Sagi and Rokach 2018). The main factors behind the ensemble system’s success are increasing diversity among baseline classifier types, using different ensemble methods, using different beginning parameters, and creating multiple datasets from the original dataset (cross-validation

or sub-samples) (Mohammed and Kora 2021). Ensemble methods aim to increase prediction accuracy by combining decisions from various sub-models into a new model. Besides, the ensemble methods help avoid overfitting and reduce variance and biases. Also, ensemble learning helps to generate multiple hypotheses using the same base learner. In addition, ensemble learning methods help reduce the drawbacks of the baseline models (Alojail and Bhatia 2020). The most popular ensemble techniques for enhancing machine learning performance are bagging, boosting, and stacking. Table 1 describes the advantages and disadvantages of each.

There are several domains using ensemble learning methods to generalize machine learning techniques, such as natural language processing (NLP), internet of things (IoT), recommender systems, face recognition, information security, information retrieval, image retrieval, and intrusion detection system (Mohammed and Kora 2021; Forouzandeh et al. 2021; Yaman et al. 2018; Pashaei Barbin et al. 2020). Also, in sentiment analysis, many research studies have shown the superiority of the different ensemble learning methods over traditional machine learning classifiers. For example, the research efforts of Kanakaraj and Gudetti (2015); Prusa et al. (2015); Wang et al. (2014); Alrehili and Albalawi (2019); Sharma et al. (2018); Fersini et al. (2014); Perikos and Hatzilygeroudis (2016); Onan et al. (2016) applied a bagging method on a several of baseline classifiers such as (NB, SVM, KNN, LR, DT, ME) for English sentiment analysis. Also, the authors in Xia et al. (2011); Tsutsumi et al. (2007); Rodriguez-Penagos et al. (2013); Clark and Wicentwoski (2013); Li et al. (2010) applied two ensemble methods by voting and stacking based on NB, SVM and LR for

**Table 1** Summary of ensemble methods

Ensemble methods	Advantage	Disadvantage
Bagging	<ul style="list-style-type: none"> <li>- Ease of implementation and adapts.</li> <li>- Reducing Variance (Avoids Overfitting).</li> <li>- High performs on high-dimensional data.</li> <li>-Allowing weak learners to outperform strong learner</li> <li>-Robust against to noise or outliers data</li> </ul>	<ul style="list-style-type: none"> <li>-High Bias</li> <li>-Computationally Expensive</li> <li>-Loss of interpretability of the model</li> </ul>
Boosting	<ul style="list-style-type: none"> <li>-Reduces Variance.</li> <li>-Reduces Bias.</li> <li>-Handling of the missing data.</li> <li>- Ease of interpretation of the model</li> </ul>	<ul style="list-style-type: none"> <li>-Slower to train</li> <li>- Computationally Expensive</li> <li>-More Overfitting</li> <li>-The difficulty of scaling sequential training</li> <li>-Each classifier must correct the errors made by its predecessors</li> </ul>
Stacking	<ul style="list-style-type: none"> <li>-A deeper understanding of the data.</li> <li>-More Accurate</li> <li>-Less Variance</li> <li>-Less Bias</li> <li>-Used to ensemble a variety of strong learners</li> </ul>	<ul style="list-style-type: none"> <li>-More Overfitting</li> <li>- Time Complexity</li> <li>-The difficulty of interpreting the final model</li> </ul>

English sentiment analysis. In addition, the authors in Da Silva et al. (2014); Xia et al. (2016); Fersini et al. (2016); Araque et al. (2017); Saleena (2018) applied majority voting based on several traditional classifiers such as SVM, RF, LR, NB, DT, and ME for English sentiment analysis. At the same time, several studies applied a stacking based on traditional classifiers for non-English sentiment analysis. For example, the authors in Lu and Tsou (2010); Li et al. (2012); Su et al. (2012) applied a stacking based on KNN, NB, SVM, and ME for Chinese reviews, the authors in Pasupulety et al. (2019) applied a stacking based on SVM and RF for India's reviews. In contrast, few studies applied ensemble learning techniques based on traditional classifiers of the Arabic language and its different dialects. For example, the authors in Saeed et al. (2022) applied a stacking based on SVM, NB, LR, DT, and KNN for Arabic sentiment analysis. But the authors in Oussous et al. (2018) applied a stacking based on SVM and ME for Moroccan tweets. On the other hand, ensemble-based deep learning models are a powerful alternative to traditional ensemble learning methods. Ensemble deep learning has shown excellent performance in sentiment analysis. For example, the researchers in Deriu et al. (2016); Akhtyamova et al. (2017) applied two ensemble methods by voting and stacking based on CNN for English sentiment analysis. Similarly, the work in Xu et al. (2016); Araque et al. (2017); Mohammadi and Shaverizade (2021); Haralabopoulos et al. (2020) applied voting and stacking based on LSTM and CNN for English sentiment analysis. However, the researchers in Heikal et al. (2018) applied voting based on CNN, GRU, and LSTM for Arabic sentiment analysis.

### 3 Proposed meta-ensemble deep learning approach

The meta-ensemble deep learning approach architecture consists of three layers, which are level-1, level-2, and level-3, as in Fig. 1. Level 1 represents the input layer, where each board of ( $M$ ) models is trained independently using a unique training dataset and different deep

architectures. Level 2 represents the meta-learner's hidden layer, in which each board model's prediction outputs in the previous layer are combined using a meta-learner. Level 3 represents the output meta-learner layer. At this level, the outputs of all predictions of the level-2 meta-learner are combined using the final level of the meta-learner to produce the final results. The proposed approach in abstract form can be seen as a general meta-neural network in which the first level is considered the input layer, level 2 is the hidden layer that acts as an activation function, and level 3 is the output layer.

#### 3.1 Description of the proposed Algorithm

The formal semantics of the proposed training procedure of the proposed approach is shown in algorithm 1. The algorithm starts by randomly generating  $N$  equally-size samples from a training dataset  $Data^{(0)}$ . Each data sample  $Data_i^{(0)} = (train_i^{(0)}, test_i^{(0)})$  is splitted into two parts; training and testing data. At the Baseline Learning procedure, the  $Level - 1$  learning models are generated by applying  $M BL_i$  Baseline Deep learners on each training dataset  $(train_i^{(0)})$ . As a result, we have  $n$  boards  $C_i, 1 \leq i \leq n$  each containing  $M$  diverse baseline models  $C_i = Model_{i1}, Model_{i2}, \dots, Model_{iM}$ . For each test,  $Test_i^{(0)} = (X^{(0)}, Y^{(0)})$ , of the  $n$  data samples are used to create metadata  $Data_i^{(1)}$  of the next level by stacking all the predicted output of each model  $Model_i$ . Each  $Data_i^{(1)}$  in level-2 has  $M + 1$  features:  $M$  features result from the prediction of the model in the board  $C_i$  on the  $test^{(0)}$ , and one extra feature represents the class label  $Y^{(0)}$ . In  $Level - 2$  once metadata has been generated, a set  $ShallowClf$  of  $n$  shallow meta classifier is used to generate the models of Level-2. Following the creation of Level-2 models,  $test_i^{(1)} = (X^{(a)}, Y^{(1)})$  are utilized to construct top the final meta data of  $Level - 3$ . Likewise the previous level, the top metadata are generated in two steps. The first step generates  $Data_i^{(1)}$  of  $n + 1$  features results from the predictions of Level-2 models on  $X^{(1)}$  and target class  $Y^{(1)}$ . In the next step, we construct  $Data_i^{(1)}$  to form the final metadata. A Final meta learner is utilized to learn those top metadata in the Level-3 learning phase.

**Procedure 1** Proposed Multi-level Training Algorithm

---

```

1: procedure GENERATING Input: Data
2:   Generate  $Data_i^{(0)} = Train_i^{(0)} \cup Test_i^{(0)} = (X_i^{(0)}, Y_i^{(0)})$ ,  $1 \leq i \leq n$ 
3: procedure BASELINE LEARNING: Level1
4:    $BaseModels = \{BL_1, BL_2, \dots, BL_M\}$  a group of  $M$  Baseline Deep learners
5:   for each  $Train_i^{(0)}$ ,  $1 \leq i \leq n$  do
6:     for each  $BL_j \in BaseModels$ ,  $1 \leq j \leq M$  do
7:        $Model_{ij} \leftarrow fit(BL_j, Train_i^{(0)})$ ,  $1 \leq j \leq M$ 
8:        $C_i \leftarrow \{Model_{i1}, Model_{i2}, \dots, Model_{iM}\}$ ,  $1 \leq i \leq n$ 
9:     for each  $Model_{ij} \in C_i$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq M$  do
10:       $y_{ij}^{(1)} \leftarrow Model_{ij}(X_i^{(0)})$ ,  $1 \leq j \leq M$ 
11:       $Data_i^{(1)} \leftarrow FeatureStacking([y_{i1}^{(1)}, y_{i2}^{(1)}, \dots, y_{ik}^{(1)}, Y_i^{(0)}])$ ,  $1 \leq i \leq n$ 
12: procedure LEARNING: LEVEL2
13:   Divide  $Data_i^{(1)} = Train_i^{(1)} \cup (Test_i^{(1)} = (X_i^{(1)}, Y_i^{(1)}))$ ,  $1 \leq i \leq n$ 
14:    $ShallowClf = \{sh_1, sh_2, \dots, sh_n\}$  a group of  $n$  shallow learners
15:   for each  $Train_i^{(1)}$ ,  $1 \leq i \leq n$  do
16:      $shModel_j \leftarrow fit(sh_i, Train_i^{(1)})$ ,  $1 \leq j \leq n$ 
17:      $shallowModels \leftarrow \{shModel_1, shModel_2, \dots, shModel_n\}$ 
18: procedure FINAL LEVEL LEARNING: LEVEL3
19:   for each  $Test_i^{(1)} = (X_i^{(1)}, Y_i^{(1)})$ ,  $1 \leq i \leq n$  do
20:     for each  $shModel_j \in shallowModels$  do
21:        $y_{ij}^{(2)} \leftarrow predict(shModel_j(X_i^{(1)})$ 
22:        $Data_i^{(2)} \leftarrow PredictedStacking[y_{i1}^{(2)}, y_{i2}^{(2)}, \dots, y_{in}^{(2)}, Y_i^{(1)}]$ ,  $1 \leq i \leq n$ 
23:        $TopMetaData = stacked([Data_1^{(2)}, Data_2^{(2)}, \dots, Data_n^{(2)}]^T)$ 
24:        $FinalClassifier \leftarrow$  is top level classifier
25:        $FinalModel \leftarrow fit(FinalClassifier, TopMetaData)$ 

```

---

## 4 Experiment results

This section describes the benchmark datasets used for sentiment analysis, the selection of baseline deep models, and shallow meta-classifiers in the framework of the proposed meta-ensemble deep learning approach scheme.

### 4.1 Description of benchmark datasets

To evaluate the extended meta-ensemble deep learning approach, we selected six sentiment benchmark datasets for conducting the experiments based on English, Arabic, and different dialects: We propose the first dataset called “Arabic-Egyptian corpus 2”, which made up of 40,000 annotated tweets from the corpus (Mohammed and Kora 2019), and another extension of 10 K tweets which is available in Kora and Mohammed (2022). The later extension consists of 5k positive and 5k negative tweets from the Arabic language and the Egyptian dialect. The second dataset includes tweets in the Saudi dialect related to distance learning during the Covid19 pandemic (Aljabri et al. 2021). It contains a total of 1675 tweets, which includes more positive tweets than negative tweets. The third dataset is ASTD (Nabil et al. 2015). It

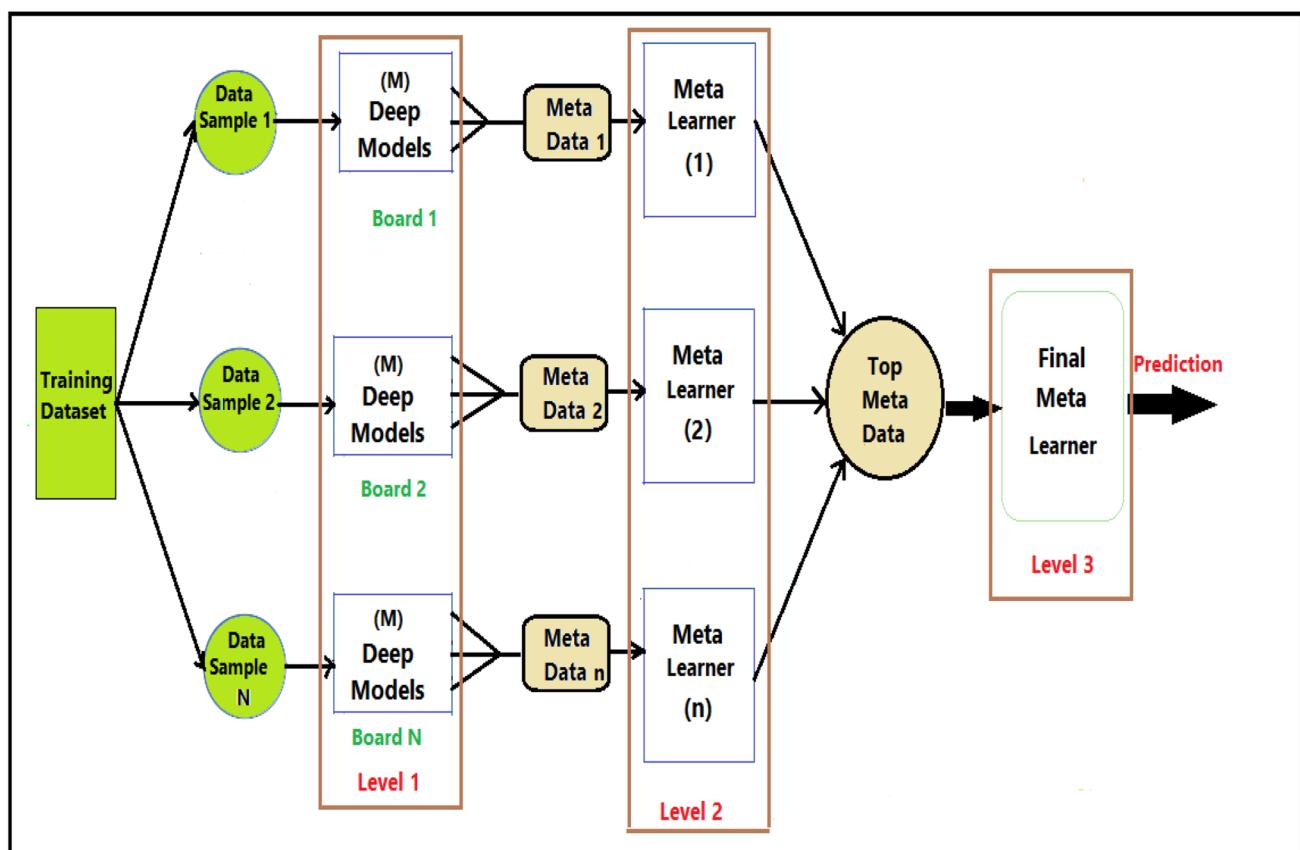
contains about 10K Arabic tweets from different dialects and is classified into 797 positive and 1682 negative (Table 2). Tweets were annotated as positive, neutral, negative, and mixed. The fourth dataset is ArSenTD-LEV (Al-Laith and Shahbaz 2021). It contains 4000 tweets from countries in the Levant Region, such as Jordan, Palestine, Lebanon and Syria. The fifth dataset is Movie Reviews (Koh et al. 2010). It contains 10,662 reviews, divided into 5331 negative and 5331 positives. The sixth dataset is the Twitter US Airline Sentiment dataset (Rane and Kumar 2018). Table 3 summarizes the characteristics of different benchmark datasets for sentiment analysis. It contains 14,600 customer tweets from six airlines in the US, including negative, positive, and neutral sentiments. In general, the textual data was preprocessed using one-hot encoding or word-embedding (Lai et al. 2016), as an initial layer before training the network. Only the positive and negative binary sentiment polarity labels are used for each dataset, and the other polarity labels are neglected. In our experiments, we divided each benchmark dataset into training and validation test sets with a ratio of (80%, 20%). In addition, we divided each benchmark dataset into eight partitions.

**Table 2** Applications of ensemble learning to sentiment classification

Approach	Papers	Baseline classifiers	Ensemble method	Languages	Dataset
TEL	Wilson et al. (2006)	DT	Boosting	English	MPQA Corpus (Wiebe et al. 2005)
	Tsutsumi et al. (2007)	SVM, ME	Stacking	English	Movie Review (Chaovalit and Zhou 2005)
	Li et al. (2010)	SVM, LR	Voting	English	Amazon.com. (Rushdi-Saleh et al. 2011)
	Lu and Tsou (2010)	NB, ME, SVM	Stacking	Chinese	Reviews (Seki et al. 2008)
	Xia et al. (2011)	NB, ME, SVM	Stacking	English	Movie Review (Chen et al. 2012)
	Li et al. (2012)	SVM, KNN	Stacking	Chinese	Reviews (Seki et al. 2008)
	Su et al. (2012)	ME, SVM	Voting, Stacking	Chinese	Reviews (Seki et al. 2008)
	Rodriguez-Penagos et al. (2013)	SVM	Voting	English	SemEval (Dzikovska et al. 2013)
	Clark and Wicentwoski (2013)	NB	Voting	English	SemEval (Nakov et al. 2016)
	Fersini et al. (2014)	ME, SVM, NB	Voting, Bagging	English	Product Reviews Pang and Lee (2005)
	Da Silva et al. (2014)	SVM, RF, LR	Voting	English	Tweets Saif et al. (2013)
	Wang et al. (2014)	SVM, KNN, DT, ME, NB	Bagging, Boosting	English	Movie Reviews (Chaovalit and Zhou 2005)
	Kanakaraj and Guddeti (2015)	NB, SVM	Bagging, Boosting	English	Movie Review (Chen et al. 2012)
	Prusa et al. (2015)	KNN, SVM, LR	Bagging, Boosting	English	sentiment140 Corpus (Go et al. 2009)
	Xia et al. (2016)	SVM, LR	Voting	English	Amazon.com. (Rushdi-Saleh et al. 2011)
	Onan et al. (2016)	BLR, NB, LDA, LR, SVM	Stacking, AdaBoost, Bagging	English	Tweets (Whitehead and Yaeger 2009)
	Fersini et al. (2016)	NB, DT, SVM	Voting	English	Movie Reviews (Chen et al. 2012)
	Perikos and Hatzilygeroudis (2016)	NB, ME	Bagging	English	Posts (Cambria et al. 2013)
	Araque et al. (2017)	NB, ME, SVM	Voting	English	Movie Reviews (Chen et al. 2012)
	Oussous et al. (2018)	MNB, SVM, ME	Voting, Stacking	Moroccan	Tweets (Tratz et al. 2013)
	Saleena (2018)	SVM, RF, NB, LR	Voting	English	Sentiment140 Corpus (Go et al. 2009), Tweets (Speriosu et al. 2011)
	Sharma et al. (2018)	SVM	Bagging	English	Movie Reviews (Chen et al. 2012)
	Pasupulety et al. (2019)	SVM, RF	Stacking	Indian	NSE (Kumar and Misra 2018)
	Saeed et al. (2022)	SVM, NB, LR, DT, KNN	Voting, Stacking	Arabic	Corpus (Li et al. 2011)

**Table 2** (continued)

Approach	Papers	Baseline classifiers	Ensemble method	Languages	Dataset
EDL	Deriu et al. (2016)	CNN	Stacking	English	SemEval (Bethard et al. 2016)
	Xu et al. (2016)	CNN, LSTM	Voting	English	SemEval (Dzikovska et al. 2013)
	Akhtyamova et al. (2017)	CNNs	Voting	English	Reviews (Karimi et al. 2015)
	(Araque et al. 2017)	CNN, LSTM, GRU	Voting, Stacking	English	Movie reviews (Chen et al. 2012)
	(Heikal et al. 2018)	CNN, LSTM	Voting	Arabic	ASTD (Nabil et al. 2015)
	Haralabopoulos et al. (2020)	LSTM, GRU, CNN, RCNN, DNN	Voting, Stacking	English	Comments (van Aken et al. 2018), SemEval (Bethard et al. 2016)
	(Mohammadi and Shaveri-zade 2021)	CNN, LSTM, GRU, Bi_LSTM	Stacking	English	SemEval (Bethard et al. 2016)

**Fig. 1** The general architecture of the proposed meta-ensemble deep learning approach

## 4.2 Baseline deep learning models

To enhance the performance of predictions in sentiment analysis through the proposed meta-ensemble deep learning

approach, we first need to build a set of deep learning models that form the baseline classifiers of the proposed meta-ensemble deep learning approach for each benchmark dataset. Three deep baseline models are proposed in this research: Long Short-Term Memory (LSTM) is the first

**Table 3** Distribution of the different benchmark dataset

Dataset	Data types	Sentiment classes	Positive count	Negative count	Total count
1-Arabic-Egyptian Corpus (Mohammed and Kora 2019; Kora and Mohammed 2022)	Egyptian dialects, MSA	2	25k	25k	50k
2-Saudi Arabia Tweets (Aljabri et al. 2021)	Dialects Tweets	2	1002	673	1675
3-ASTD (Nabil et al. 2015)	Dialects Tweets	4	797	1682	10,006
4-ArSenTD-LEV (Al-Laith and Shahbaz 2021)	Dialects Tweets	5	835	1253	4,000
5-Movie Reviews (Koh et al. 2010)	English Reviews	2	5331	5331	10,662
6-Twitter US Airline Sentiment (Rane and Kumar 2018)	English Tweets	3	2310	8797	14,601

**Table 4** Configurations of baseline deep learning models

Models	Configuration value
GRU	GRU layer= 1 or 2
	GRU size= 256
LSTM	LSTM layer= 1 or 2
	LSTM size= 256
CNN	No. of filters= 32
	Filters size= 16
	Vocab size= 10,000

baseline deep model utilized in our evaluation (Mohammed and Kora 2019). The LSTM model is a well-known architecture for representing sequential data. It was designed better to capture long-term dependencies than the recurrent neural network model. Three gates comprise LSTM architecture: the input gate, the forget gate, and the output gate. The Gated recurrent unit (GRU) is the next baseline deep model (Pan et al. 2020). The GRU model is comparable to the LSTM model, except it contains fewer parameters. GRU comprises of two gates: the reset gate and the update gate. The Convolutional Neural Network Model (CNN) is the third baseline deep model (Abdulnabi et al. 2015). The CNN model is a feedforward neural network consisting of one or more convolutional layers and a fully connected layer, which also includes a pooling layer for integration. In general, each deep baseline model is trained on different hyperparameters. Table 4 shows the configurations of baseline deep learning models. Table 5 shows the accuracy of each data split within each dataset and the average accuracy of each baseline deep model in each dataset. It should be mentioned that the experimental results reveal that the highest average accuracy obtained in the first dataset of Arabic-Egyptian Corpus is 89.38% of the LSTM model. Also, the highest average accuracy obtained in the second dataset of Saudi Arabia Tweets is 65.38% of the LSTM2 model. In addition, the highest average accuracy obtained in the third ASTD dataset is 71.6% of the LSTM model. Moreover, the highest average accuracy obtained in the fourth ArSenTD-LEV dataset is

76.2% of the LSTM model. Additionally, the highest average accuracy obtained in the fifth dataset of the Movie Reviews dataset is 78.03% of the LSTM1 model. Finally, the highest average accuracy obtained in the Twitter US Airline Sentiment dataset's sixth dataset is 80.05% of the LSTM1 model. In the conducted experiments, 114 deep baseline models in all have been trained. In addition, the sizes of the baseline models vary on each dataset. In Saudi Arabia, tweets, Movie Reviews, and Twitter US Airline Sentiment are 4 deep baseline models, while ASTD and ArSenTD-LEV are 3 deep baseline models.

### 4.3 Meta-ensemble classifiers

To combine the trained baseline deep models within the boards of models, we use a set of shallow meta-classifiers that include Support Vector Machines (SVM), Gradient Boosting (GB), Naive Bayes (NB), Random Forest (RF), Logistic Regression (LG) as top surface meta learners. Table 6 describes the accuracy results of the proposed clustering method in each dataset. In the first dataset of Arabic-Egyptian Corpus, the results indicate that the ensemble with SVM classifier achieved the best accuracy in both hard and soft prediction with a score of 92.6% and 93.2%, respectively. In the second dataset of Saudi Arabian tweets, the results indicate that the ensemble with the SVM classifier achieved the best accuracy in the hard prediction of 69.9%. In contrast, the ensemble with both the SVM and LG classifier achieved the best soft prediction accuracy with a score of 72.3%. In the third dataset of ASTD, the results indicate that both the ensemble with SVM and LG classifier achieved the best accuracy in hard prediction with a score of 75.9%. At the same time, the ensemble with the LG classifier achieved the best accuracy in soft prediction with a score of 77.6%. In the fourth dataset of ArSenTD-LEV, the results indicate that the ensemble with the SVM classifier achieved the best accuracy in hard prediction with a score of 80.4%. In contrast, the ensemble with the LG classifier achieved the best accuracy in soft prediction with a score of 83.2%. In

**Table 5** Performance accuracy results of baseline deep classifiers in different datasets

Dataset	Baseline models	Split dataset								<b>AVG models (%)</b>
		1 (%)	2 (%)	3 (%)	4 (%)	5 (%)	6 (%)	7 (%)	8 (%)	
1-Mohammed and Kora (2019); Kora and Mohammed (2022)	GRU	<b>89.9</b>	<b>89.8</b>	89.2	<b>89.5</b>	<b>89.3</b>	88.8	88.2	89	89.21
	LSTM	89.7	<b>89.8</b>	<b>89.8</b>	89.1	89.2	<b>89</b>	<b>89.1</b>	<b>89.4</b>	<b>89.38</b>
	CNN	87.64	85.04	84.78	85.65	87.40	86	85	86.2	85.96
2-Aljabri et al. (2021)	GRU1	<b>63.1</b>	67.3	<b>66.2</b>	64.2	62.3	64.2	<b>65.4</b>	67.7	65.05
	LSTM1	61.9	61.9	60	65	64.6	68.5	65	<b>70.8</b>	64.71
	GRU2	60.8	<b>69.6</b>	60.4	61.5	63.1	66.9	64.6	61.5	63.55
	LSTM2	60.4	65.4	65	<b>66.2</b>	<b>66.5</b>	<b>70</b>	62.3	67.3	<b>65.38</b>
	CNN	—	—	—	—	—	—	—	—	—
3-Nabil et al. (2015)	GRU1	<b>73.1</b>	66.2	68.5	72.8	<b>74.1</b>	<b>72.3</b>	<b>72.8</b>	67.9	70.86
	LSTM1	72.1	<b>75.9</b>	<b>69.5</b>	<b>74.1</b>	71.5	69.2	69	<b>71.5</b>	<b>71.6</b>
	GRU2	—	—	—	—	—	—	—	—	—
	LSTM2	—	—	—	—	—	—	—	—	—
	CNN	68.2	70.4	67	68	68.9	71	70.6	68.2	69.03
4-Al-Laith and Shahbaz (2021)	GRU1	<b>74.5</b>	<b>76.4</b>	73.6	73.9	76.4	77.3	<b>78.5</b>	76.4	75.87
	LSTM1	73.3	75.8	<b>75.2</b>	<b>78.2</b>	75.2	<b>78.2</b>	75.8	<b>77.9</b>	<b>76.2</b>
	GRU2	—	—	—	—	—	—	—	—	—
	LSTM2	—	—	—	—	—	—	—	—	—
	CNN	70.5	76	65.5	75	71.3	77.3	75.5	66	72.13
5-Koh et al. (2010)	GRU1	68.9	<b>77.5</b>	76.6	75.4	71.2	75.3	74.8	<b>76.6</b>	74.53
	LSTM1	<b>82.6</b>	74.8	<b>79.4</b>	<b>81.9</b>	<b>81.7</b>	<b>76.4</b>	<b>82.7</b>	64.8	<b>78.03</b>
	GRU2	62.4	57.4	55.4	64.1	66.1	58.2	69	69.2	62.72
	LSTM2	71.9	67.9	62.4	68.4	54.8	66.8	65.9	74.9	66.62
	CNN	—	—	—	—	—	—	—	—	—
6-Rane and Kumar (2018)	GRU1	71.6	<b>78.6</b>	79.2	78.9	68.5	70.4	65.9	73.3	73.18
	LSTM1	<b>80.6</b>	78.4	<b>81.1</b>	<b>79.7</b>	<b>80.3</b>	<b>81.8</b>	<b>78.1</b>	<b>81.2</b>	<b>80.05</b>
	GRU2	70.6	66.4	70.8	68.2	63.3	67.7	64.4	63.9	66.82
	LSTM2	73.2	66.3	72.4	69.7	71.3	70.8	70.9	73.1	70.96
	CNN	—	—	—	—	—	—	—	—	—

The values in bold indicate superior results among the baseline models in each data split

**Table 6** Performance Accuracy of the proposed Meta-Ensemble in different datasets

Dataset	Predictions	GB (%)	SVM (%)	NB (%)	LG (%)	RF (%)
1-Mohammed and Kora (2019); Kora and Mohammed (2022)	Hard	92	<b>92.6</b>	91.6	91.9	91.9
	Soft	91.8	<b>93.2</b>	92.2	92.3	90
2-Aljabri et al. (2021)	Hard	69.3	<b>69.9</b>	67.4	69.2	68.4
	Soft	71.2	<b>72.3</b>	69.8	<b>72.3</b>	71.8
3-Nabil et al. (2015)	Hard	74.1	<b>75.9</b>	72.3	<b>75.9</b>	74.1
	Soft	76.2	77.1	73.6	<b>77.6</b>	75.8
4-Al-Laith and Shahbaz (2021)	Hard	79.5	<b>80.4</b>	76.2	80.3	79.6
	Soft	81.4	82.3	79.1	<b>83.2</b>	81.4
5-Koh et al. (2010)	Hard	80.5	<b>80.9</b>	79.3	80.5	80.5
	Soft	82.4	<b>83.9</b>	80.5	83.8	82.1
6-Rane and Kumar (2018)	Hard	82.1	<b>82.9</b>	80.3	81.8	82.2
	Soft	<b>85.3</b>	85.1	81.9	85.1	84.9

The values in bold indicate superior results among the baseline models in each data split

**Table 7** Summary of accuracy

Benchmarks	AVG Baseline models	High AVG Baseline models	Meta-Ensemble
1-Mohammed and Kora (2019); Kora and Mohammed (2022)	GRU= 89.52% LSTM= 89.54% CNN= 86.10%	LSTM= 89.54%	SVM= <b>93.2%</b> (Soft)
2-Aljabri et al. (2021)	GRU1= 65.05% LSTM1= 64.71% GRU2= 63.55% LSTM2= 65.38%	LSTM2= 65.38%	SVM= <b>72.3%</b> (Soft)
3-Nabil et al. (2015)	GRU= 70.86% LSTM= 71.6% CNN= 69.03%	LSTM= 71.6%	LG= <b>77.6%</b> (Soft)
4-Al-Laith and Shahbaz (2021)	GRU= 75.87% LSTM= 76.2% CNN= 72.13%	LSTM= 76.2%	LG= <b>83.2%</b> (Soft)
5-Koh et al. (2010)	GRU1= 74.53% LSTM1= 78.03% GRU2= 62.72% LSTM2= 66.62%	LSTM1= 78.03%	SVM= <b>83.9%</b> (Soft)
6-Rane and Kumar (2018)	GRU1= 73.18% LSTM1= 80.05% GRU2= 66.82% LSTM2= 70.96%	LSTM1= 80.05%	GB= <b>85.3%</b> (Soft)

The values in bold indicate superior results among the meta classifiers in each data split

the fifth Movie Reviews dataset, the results indicate that the ensemble with the SVM classifier achieved the best accuracy in both hard and soft prediction with a score of 80.9% and 83.9%, respectively. In the sixth dataset of Twitter US Airline Sentiment, the results indicate that the ensemble with the SVM classifier achieved the best accuracy in hard prediction with a score of 82.9%. At the same time, the ensemble with the GB classifier achieved the best accuracy in soft prediction with a score of 85.3%. Table 7 compares the highest accuracy results of the average baseline deep models with the highest accuracy results of meta-ensemble classifiers in each dataset. It can be noted that the highest average accuracy was obtained in the proposed meta-ensemble in the different datasets in soft prediction. Also, it can be noted that the highest average accuracy obtained in baseline deep models in the different datasets is the LSTM model than in the other networks. In general, it can be noted that different meta-ensemble classifiers show better performance for the final prediction. It can also be noted that using 5-fold cross-validation on the predictions of deep baseline models, SVM is shown as the most frequent best combiner to fuse the boards of models in the level-1 with 93.2%, 72.3% and 83.9% in each of the Arabic-Egyptian Corpus, Saudi Arabia Tweets and Movie

Reviews datasets, respectively. In addition, LG is shown as the most frequent best combiner to fuse the boards of models in level-1 with 77.6% and 83.2% in both the ASTD and ArSenTD-LEV datasets, respectively. Finally, GB is considered the most frequent best combiner to fuse the models' boards in the level-1 at 85.3% in the Twitter US Airline Sentiment datasets.

## 5 Conclusion

Deep learning models have shown great success in sentiment analysis in the literature. However, modeling an effective deep learning model requires great effort due to finding the best architecture of the neural network and the best configuration of hyperparameters. An approach for tackling these limitations is using the ensemble methods. The key idea of the ensemble is to produce a powerful learner using a combination of weak learners. Thus, in this research paper, we proposed a meta-ensemble deep learning approach to improve the performance of sentiment analysis. This proposed approach combines the predictions of several groups of deep models using three levels of the meta-learning method. Also, we proposed the benchmark dataset “Arabic-Egyptian

Corpus 2". This corpus comprises 10,000 annotated tweets written in colloquial Arabic on various topics. This corpus is added to the original version in Mohammed and Kora (2019) that contains 40K annotated tweets. We conducted several experiments on six public benchmark datasets for sentiment analysis involving several languages and dialects to test and evaluate the performance of the proposed meta-ensemble deep learning approach. We trained sets of baseline classifiers (GRU, LSTM, and CNN) on each benchmark dataset, and their best model was compared with the proposed meta-ensemble deep learning approach. In particular, we have trained 114 deep models and performed a comparison on five different shallow meta-classifiers to ensemble those models. The experimental results revealed that the meta-ensemble deep learning approach effectively outperforms all six benchmark datasets' baseline deep learning models. Also, the experiments suggested that the meta-learners work better when the predictions of the involved layers are of the form probability distribution. In summary, the proposed ensemble approach uses parallel ensemble techniques where baseline learners are generated simultaneously, as there is no data dependency and the fusion methods depend on the meta-learning method. However, our proposed approach has some challenges and limitations, such as determining the appropriate number of baseline models and selecting baseline models that can be relied upon to generate the best predictions from each dataset when designing our meta-ensemble deep learning approach from scratch. Also, the difficulty of computing time complexity is added when the amount of available data grows exponentially. In addition, the issue of multi-label classification raises many problems, such as overfitting and the curse of dimensionality, in the case of high dimensionality of data. Handling a multi-class problems worth investigating in case of multi-level ensemble. Also, transformer models recently received more attention in NLP tasks. It is worth investigating the impact of ensemble learning with transformers with full extensive experiments.

**Author contributions** Paper is written by AM and RK Paper is reviewed by AM.

**Funding** Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are

included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Abdulnabi AH, Wang G, Lu J, Jia K (2015) Multi-task cnn model for attribute prediction. *IEEE Trans Multimedia* 17(11):1949–1959
- Ahmed S, Pasquier M, Qadah G (2013) Key issues in conducting sentiment analysis on arabic social media text. In: 2013 9th International conference on innovations in information technology (IIT), pp 72–77. IEEE
- van Aken B, Risch J, Krestel R, Löser (2018) A challenges for toxic comment classification: an in-depth error analysis. In: ALW
- Akhtyamova L, Ignatov A, Cardiff J (2017) A large-scale cnn ensemble for medication safety analysis. In: International conference on applications of natural language to information systems, pp 247–253. Springer
- Al-Laith A, Shahbaz M (2021) Tracking sentiment towards news entities from arabic news on social media. *Future Gener Comput Syst* 118:467–484
- Aljabri M, Chrouf SMB, Alzahrani NA, Alghamdi L, Alfehaid R, Alqarawi R, Alhuthayfi J, Alduhailan N (2021) Sentiment analysis of arabic tweets regarding distance learning in saudi arabia during the covid-19 pandemic. *Sensors* 21(16):5431
- Alojail M, Bhatia S (2020) A novel technique for behavioral analytics using ensemble learning algorithms in e-commerce. *IEEE Access* 8:150072–150080
- Alomari KM, ElSherif HM, Shaalan K (2017) Arabic tweets sentimental analysis using machine learning. In: International conference on industrial, engineering and other applications of applied intelligent systems, pp 602–610. Springer
- Alrehili A, Albalawi K (2019) Sentiment analysis of customer reviews using ensemble method, pp 1–6
- Araque O, Corcuera-Platas I, Sánchez-Rada JF, Iglesias CA (2017) Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Syst Appl* 77:236–246
- Baly R, El-Khoury G, Moukalled R, Aoun R, Hajj H, Shaban KB, El-Hajj W (2017) Comparative evaluation of sentiment analysis methods across arabic dialects. *Procedia Comput Sci* 117:266–273
- Bethard S, Savova G, Chen WT, Derczynski L, Pustejovsky J, Verhagen M (2016) SemEval-2016 task 12: clinical tempeval. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp 1052–1062
- Cambria E, Das D, Bandyopadhyay S, Feraco A, et al (2017) A practical guide to sentiment analysis
- Cambria E, Schuller B, Xia Y, Havasi C (2013) New avenues in opinion mining and sentiment analysis. *IEEE Intell Syst* 28(2):15–21
- Chan S, Reddy V, Myers B, Thibodeaux Q, Brownstone N, Liao W (2020) Machine learning in dermatology: current applications, opportunities, and limitations. *Dermatol Therapy* 10(3):365–386
- Chaovalit P, Zhou L (2005) Movie review mining: a comparison between supervised and unsupervised classification approaches. In: Proceedings of the 38th annual Hawaii international conference on system sciences, pp 112c–112c. IEEE
- Chen L, Wang W, Nagarajan M, Wang S, Sheth A (2012) Extracting diverse sentiment expressions with target-dependent polarity from twitter. In: Proceedings of the international AAAI conference on web and social media, vol 6, pp 50–57

- Chen Y, Yuan J, You Q, Luo J (2018) Twitter sentiment analysis via bi-sense emoji embedding and attention-based lstm. In: 2018 ACM Multimedia conference on multimedia conference, pp 117–125. ACM
- Cho SB, Won HH (2003) Machine learning in dna microarray analysis for cancer classification. In: Proceedings of the First Asia-Pacific bioinformatics conference on bioinformatics 2003-volume 19, pp 189–198
- Clark S, Wicentwoski R (2013) Swates: combining simple classifiers with estimated accuracy. In: Second joint conference on lexical and computational semantics (\* SEM), volume 2: proceedings of the seventh international workshop on semantic evaluation (SemEval 2013), pp 425–429
- Collobert R, Weston J (2008) A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th international conference on machine learning, pp 160–167
- Da Silva NF, Hruschka ER, Hruschka ER Jr (2014) Tweet sentiment analysis with classifier ensembles. *Decis Support Syst* 66:170–179
- Deriu J, Gonzenbach M, Uzdilli F, Lucchi A, Luca VD, Jaggi M (2016) Swisscheese at semeval-2016 task 4: sentiment classification using an ensemble of convolutional neural networks with distant supervision. In: Proceedings of the 10th international workshop on semantic evaluation, CONF, pp 1124–1128
- Duwairi RM, Marji R, Sha'ban N, Rushaidat S (2014) Sentiment analysis in arabic tweets. In: 2014 5th International conference on information and communication systems (ICICS), pp 1–6. IEEE
- Dzikovska MO, Nielsen RD, Brew C, Leacock C, Giampiccolo D, Bentivogli L, Clark P, Dagan I, Dang HT (2013) Semeval-2013 task 7: the joint student response analysis and 8th recognizing textual entailment challenge. North Texas State Univ Denton, Tech. rep
- Fersini E, Messina E, Pozzi FA (2014) Sentiment analysis: Bayesian ensemble learning. *Decis Support Syst* 68:26–38
- Fersini E, Messina E, Pozzi FA (2016) Expressive signals in social media languages to improve polarity detection. *Inf Process Manag* 52(1):20–35
- Forouzandeh S, Berahmand K, Rostami M (2021) Presentation of a recommender system with ensemble learning and graph embedding: a case on movielens. *Multimedia Tools Appl* 80(5):7805–7832
- Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. CS224N project report, Stanford 1(12), 2009
- Graves A (2012) Long short-term memory. Supervised sequence labeling with recurrent neural networks, pp 37–45
- Habimana O, Li Y, Li R, Gu X, Yu G (2020) Sentiment analysis using deep learning approaches: an overview. *Sci China Inf Sci* 63(1):1–36
- Haralabopoulos G, Anagnostopoulos I, McAuley D (2020) Ensemble deep learning for multilabel binary classification of user-generated content. *Algorithms* 13(4):83
- Heikal M, Torki M, El-Makky N (2018) Sentiment analysis of arabic tweets using deep learning. *Procedia Comput Sci* 142:114–122
- Kanakaraj M, Guddeti RMR (2015) Performance analysis of ensemble methods on twitter sentiment analysis using nlp techniques. In: Proceedings of the 2015 IEEE 9th international conference on semantic computing (IEEE ICSC 2015), pp 169–170. IEEE
- Karimi S, Metke-Jimenez A, Kemp M, Wang C (2015) Cadec: a corpus of adverse drug event annotations. *J Biomed Inform* 55:73–81
- Koh NS, Hu N, Clemons EK (2010) Do online reviews reflect a product's true perceived quality? An investigation of online movie reviews across cultures. *Electron Commer Res Appl* 9(5):374–385
- Kora R, Mohammed A (2022) Arabic-Egyptian Corpus 2. <https://doi.org/10.7910/DVN/UPGJCV>
- Kulkarni NH, Srinivasan G, Sagar B, Cauvery N (2018) Improving crop productivity through a crop recommendation system using ensembling technique. In: 2018 3rd International conference on computational systems and information technology for sustainable solutions (CSITSS), pp 114–119. IEEE
- Kumar G, Misra AK (2018) Commonality in liquidity: evidence from India's national stock exchange. *J Asian Econ* 59:1–15
- Kumar V, Aydav PSS, Minz S (2021) Multi-view ensemble learning using multi-objective particle swarm optimization for high dimensional data classification. *J King Saud Univ-Comput Inf Sci*
- Lai S, Liu K, He S, Zhao J (2016) How to generate a good word embedding. *IEEE Intell Syst* 31(6):5–14
- Le NQK, Yapp EKY, Yeh HY (2019) Et-gru: using multi-layer gated recurrent units to identify electron transport proteins. *BMC Bioinform* 20(1):1–12
- Li FH, Huang M, Yang Y, Zhu X (2011) Learning to identify review spam. In: Twenty-second international joint conference on artificial intelligence
- Li S, Lee SY, Chen Y, Huang CR, Zhou G (2010) Sentiment classification and polarity shifting. In: Proceedings of the 23rd international conference on computational linguistics (Coling 2010), pp 635–643
- Li W, Wang W, Chen Y (2012) Heterogeneous ensemble learning for Chinese sentiment classification. *J Inf Comput Sci* 9(15):4551–4558
- Lu B, Tsou BK (2010) Combining a large sentiment lexicon and machine learning for subjectivity classification. In: 2010 international conference on machine learning and cybernetics, vol 6, pp 3311–3316. IEEE
- Mejova Y (2009) Sentiment analysis: an overview. University of Iowa, Computer Science Department
- Mohammadi A, Shaverzade A (2021) Ensemble deep learning for aspect-based sentiment analysis. *Int J Nonlinear Anal Appl* 12(Special Issue):29–38
- Mohammed A, Kora R (2019) Deep learning approaches for arabic sentiment analysis. *Soc Netw Anal Min* 9(1):52
- Mohammed A, Kora R (2021) An effective ensemble deep learning framework for text classification. *J King Saud Univ-Comput Inf Sci*
- Moitra D, Mandal RK (2019) Automated ajcc staging of non-small cell lung cancer (nsclc) using deep convolutional neural network (cnn) and recurrent neural network (rnn). *Health Inf Sci Syst* 7(1):1–12
- Nabil M, Aly M, Atiya A (2015) Astd: Arabic sentiment tweets dataset. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 2515–2519
- Nakov P, Rosenthal S, Kiritchenko S, Mohammad SM, Kozareva Z, Ritter A, Stoyanov V, Zhu X (2016) Developing a successful semeval task in sentiment analysis of twitter and other social media texts. *Lang Resour Eval* 50(1):35–65
- Naresh A, Venkata Krishna P (2021) An efficient approach for sentiment analysis using machine learning algorithm. *Evol Intel* 14(2):725–731
- Onan A, Korukoğlu S, Bulut H (2016) A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Syst Appl* 62:1–16
- Oussous A, Lahcen AA, Belfkikh S (2018) Improving sentiment analysis of moroccan tweets using ensemble learning. In: International conference on big data, cloud and applications, pp 91–104. Springer
- Pan M, Zhou H, Cao J, Liu Y, Hao J, Li S, Chen CH (2020) Water level prediction model based on gru and cnn. *IEEE Access* 8:60090–60100
- Pang B, Lee L (2005) Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: ACL

- Pashaei Barbin J, Yousefi S, Masoumi B (2020) Efficient service recommendation using ensemble learning in the internet of things (iot). *J Ambient Intell Humaniz Comput* 11(3):1339–1350
- Pasupuleti U, Anees AA, Anmol S, Mohan BR (2019) Predicting stock prices using ensemble learning and sentiment analysis. In: 2019 IEEE second international conference on artificial intelligence and knowledge engineering (AIKE), pp 215–222. IEEE
- Perikos I, Hatzilygeroudis I (2016) Recognizing emotions in text using ensemble of classifiers. *Eng Appl Artif Intell* 51:191–201
- Pontiki M, Galanis D, Papageorgiou H, Androutsopoulos I, Manandhar S, Mohammad AS, Al-Ayyoub M, Zhao Y, Qin B, De Clercq O, et al (2016) SemEval-2016 task 5: aspect based sentiment analysis. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp 19–30
- Prusa J, Khoshgoftaar TM, Dittman DJ (2015) Using ensemble learners to improve classifier performance on tweet sentiment data. In: 2015 IEEE international conference on information reuse and integration, pp 252–257. IEEE
- Rane A, Kumar A (2018) Sentiment classification system of twitter data for us airline service analysis. In: 2018 IEEE 42nd annual computer software and applications conference (COMPSAC), vol 1, pp 769–773. IEEE
- Rodriguez-Penagos C, Atserias J, Codina-Filba J, García-Narbona D, Grivolla J, Lambert P, Saurí R (2013) Fbm: combining lexicon-based ml and heuristics for social media polarities. In: Second joint conference on lexical and computational semantics (\*SEM), volume 2: proceedings of the seventh international workshop on semantic evaluation (SemEval 2013), pp 483–489
- Rojas-Barahona LM (2016) Deep learning for sentiment analysis. *Lang Linguist Compass* 10(12):701–719
- Rushdi-Saleh M, Martín-Valdivia MT, Ureña-López LA, Pereira-Ortega JM (2011) Oca: opinion corpus for arabic. *J Am Soc Inform Sci Technol* 62(10):2045–2054
- Saeed RM, Rady S, Gharib TF (2022) An ensemble approach for spam detection in arabic opinion texts. *J King Saud Univ-Comput Inf Sci* 34(1):1407–1416
- Sagi O, Rokach L (2018) Ensemble learning: a survey. *Wiley Interdiscip Rev: Data Min Knowl Discovery* 8(4):e1249
- Saif H, Fernandez M, He Y, Alani H (2013) Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold
- Saleena N et al (2018) An ensemble classification system for twitter sentiment analysis. *Procedia Comput Sci* 132:937–946
- Seki Y, Evans DK, Ku LW, 0001, L.S., Chen HH, Kando N (2008) Overview of multilingual opinion analysis task at ntcir-7. In: NTCIR, pp 185–203. Citeseer
- Shahzad RK, Lavesson N (2013) Comparative analysis of voting schemes for ensemble-based malware detection. *J Wirel Mobile Netw Ubiquitous Comput Depend Appl* 4(1):98–117
- Sharma S, Srivastava S, Kumar A, Dangi A (2018) Multi-class sentiment analysis comparison using support vector machine (svm) and bagging technique—an ensemble method. In: 2018 International conference on smart computing and electronic enterprise (ICSC-CEE), pp 1–6. IEEE
- Shipp CA, Kuncheva LI (2002) Relationships between combination methods and measures of diversity in combining classifiers. *Inf Fusion* 3(2):135–148
- Shoukry A, Rafea A (2012) Sentence-level arabic sentiment analysis. In: 2012 International conference on collaboration technologies and systems (CTS), pp 546–550. IEEE
- Speriosu M, Sudan N, Upadhyay S, Baldridge J (2011) Twitter polarity classification with label propagation over lexical links and the follower graph. In: Proceedings of the first workshop on unsupervised learning in NLP, pp 53–63
- Stamatatos E, Widmer G (2002) Music performer recognition using an ensemble of simple classifiers. In: ECAI, pp 335–339
- Su Y, Zhang Y, Ji D, Wang Y, Wu H (2012) Ensemble learning for sentiment classification. In: Workshop on Chinese lexical semantics, pp 84–93. Springer
- Tan KL, Lee CP, Lim KM, Anbananthen KSM (2022) Sentiment analysis with ensemble hybrid deep learning model. *IEEE Access* 10:103694–103704
- Tasci E, Uluturk C, Ugur A (2021) A voting-based ensemble deep learning method focusing on image augmentation and preprocessing variations for tuberculosis detection. *Neural Comput Appl*, pp 1–15
- Tratz S, Briesch D, Laoudi J, Voss C (2013) Tweet conversation annotation tool with a focus on an arabic dialect, moroccan darija. In: Proceedings of the 7th linguistic annotation workshop and interoperability with discourse, pp 135–139
- Tsutsumi K, Shimada K, Endo T (2007) Movie review classification based on a multiple classifier. In: Proceedings of the 21st pacific Asia conference on language, information and computation, pp 481–488
- Tuysuzoglu G, Birant D, Pala A (2018) Ensemble methods in environmental data mining. *Sch Environ Sci*, pp 1–16
- Wagh R, Punde P (2018) Survey on sentiment analysis using twitter dataset. In: 2018 Second international conference on electronics, communication and aerospace technology (ICECA), pp 208–211. IEEE
- Wang G, Sun J, Ma J, Xu K, Gu J (2014) Sentiment classification: the contribution of ensemble learning. *Decis Support Syst* 57:77–93
- Wang XY, Zhang BB, Yang HY (2013) Active svm-based relevance feedback using multiple classifiers ensemble and features reweighting. *Eng Appl Artif Intell* 26(1):368–381
- Whitehead M, Yaeger L (2009) Building a general purpose cross-domain sentiment mining model. In: 2009 WRI world congress on computer science and information engineering, vol 4, pp 472–476. IEEE
- Wiebe J, Wilson T, Cardie C (2005) Annotating expressions of opinions and emotions in language. *Lang Resour Eval* 39(2):165–210
- Wilson T, Wiebe J, Hwa R (2006) Recognizing strong and weak opinion clauses. *Comput Intell* 22(2):73–99
- Xia R, Xu F, Yu J, Qi Y, Cambria E (2016) Polarity shift detection, elimination and ensemble: a three-stage model for document-level sentiment analysis. *Inf Process Manag* 52(1):36–45
- Xia R, Zong C, Li S (2011) Ensemble of feature sets and classification algorithms for sentiment classification. *Inf Sci* 181(6):1138–1152
- Xu S, Liang H, Baldwin T (2016) Unimelb at semeval-2016 tasks 4a and 4b: An ensemble of neural networks and a word2vec based model for sentiment classification. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp 183–189
- Yadav A, Vishwakarma DK (2020) Sentiment analysis using deep learning architectures: a review. *Artif Intell Rev* 53(6):4335–4385
- Yaman MA, Subasi A, Rattay F (2018) Comparison of random subspace and voting ensemble machine learning methods for face recognition. *Symmetry* 10(11):651
- Zhang C, Ma Y (2012) Ensemble machine learning: methods and applications. Springer

## Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”). Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

[onlineservice@springernature.com](mailto:onlineservice@springernature.com)

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/368513976>

# ANGELIA: An Emotional AI for Electronic Music

Preprint · February 2023

DOI: 10.13140/RG.2.2.25416.39682

---

CITATIONS

0

READS

33

1 author:



Jean-Claude Heudin

Artificial-Creature.com

69 PUBLICATIONS 198 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Life Cellular Automata [View project](#)



KIM (Knowledge-based Integrated Machine) [View project](#)

# ANGELIA: An Emotional AI for Electronic Music

Jean-Claude Heudin

Artificial-Creature.com

Saint-Malo, France

jcheudin@artificial-creature.com

## Abstract

This paper describes the principles of ANGELIA, an Art and Artificial Intelligence project for Electronic Music in the framework of the Hyperorchestration approach. ANGELIA is a hybrid emotional AI based on a dedicated music programming language that enables to use bio-inspired algorithms for composing and performing, such as neural networks, cellular automata, fractal development, and a corpus-based genetic algorithm. It includes also a feedback loop based on an “emotional metabolism” that modifies the expressiveness of the interpretation.

## Keywords

Electronic Music, Artificial Intelligence, Bio-inspired Algorithms, Emotional Metabolism, Modular Synthesizer, Hyperinstrument, Hyperorchestration

## Introduction

Most people and even musicians have a difficult time admitting that music can be represented and understood with algorithms. However, the history of music shows that algorithms and formal approaches have played an important role in the 20th century, with composers such as Stockhausen [1], Xenakis [2], Cage [3], to cite a few, but also before with Bach [4] and Mozart [5], among others.

Nowadays computer music is an active research field encompassing a wide range of approaches [6]. The works of David Cope with EMI (Experiments in Musical Intelligence) have shown that Artificial Intelligence can be successfully used for composing music in the style of prestigious composers [7]. More recently, the advances in Deep Learning have resulted in an increasing number of studies for music generation [8].

ANGELIA is an Artificial Intelligence research project for Electronic Music. This name is the contraction of “Angel” and “IA”, the French acronym for Artificial Intelligence. The project was initiated during the summer of 2018 and its development has continued ever since.

An important axiom of ANGELIA is to place the artist at the center. Too often, the role of human is simply forgotten in AI projects, or at least unspecified. In contrast, our goal is to use AI for enhancing the creativity of the artist and not seeking to implicitly replace him. It is therefore imperative to integrate the AI in the artist’s creative work-

flow, even if it will stimulate him to reconsider his approach towards composition and orchestration.

After four years of development, this paper describes ANGELIA’s principles and approach. It first describes its hybrid architecture and the music-oriented programming language on which it is based. Then, it presents one of the main bio-inspired algorithms that can be used for composing and performing. A distinctive feature of ANGELIA is its emotional feedback loop. The next section describes the model of the “emotional metabolism” that modifies the expressiveness of the interpretation. During live performances, ANGELIA controls a dedicated 32-voice Modular Synthesizer. Together, they can be considered as a “hyper-instrument.” The last section introduces the related Hyperorchestration approach.

## Architecture Overview

ANGELIA is not based on a single algorithmic approach, such as Deep Neural Networks. Instead, it favors a hybrid architecture that enables to use different algorithms integrated in a dedicated high-level programming language.

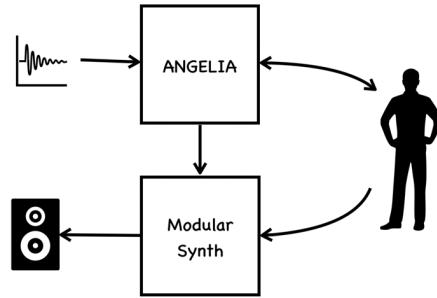


Figure 1. Overall architecture. During performances, the AI runs on a last generation tablet. The synthesizer is a 32-voice Eurorack Modular Synthesizer controlled via Midi and Control Voltage interfaces.

The architecture relies on three main parts: the artist, the AI, and the instrument. The artist interacts directly with both the AI and the electronic instrument. The AI is composed of a music generation engine and an emotional feedback loop whose purpose is to modify the expressiveness of the interpretation. The AI and the electronic instrument form together what we called a Hyperinstrument.

In fact, the boundaries between such an instrument and the instrumentalist are also subject to interpretation. We can consider them together as a hybrid being, a sort of “musical cyborg” composed of an organic part, the instrumentalist, and a machine part, the hyperinstrument, even if they are not physically merged. The concept of “musical cyborg” relies on the rejection of rigid boundaries, notably those separating human from machine [9].

## Music Programming Language

The interaction between the AI and the artist is based on a dedicated programming language and a performance-oriented interface.

Music has its own language, in the form of music scores known to all. Musicians have proposed more graphic or flexible alternatives, like Iannis Xenakis [10] or Brian Eno [11] among others. But these representations are not directly adapted for algorithmic processing. In contrast, ANGELIA is based on a music programming language that is both understandable by the composer and interpretable by the AI. This approach also allows “live coding,” a musical trend that has emerged in recent years where one can play live music with computer code [12].

The language is build on top of *JavaScript*, with an easy access to the source code, even in real-time. This code uses only one API to reduce dependencies: *WebMidi* for controlling Midi instruments [13]. The language syntax of ANGELIA inherits from trackers [14], but with a higher level of abstraction and algorithmic features. Here is an example of the language’s syntax to show its main principles:

```
! Inhumane Etude #1
SONG: Prometheus
BPM: 120
BAR: 4
UNIT: 4
LENGTH: 90

INSTRUMENT: Piano "AUM" channel:1 vcurve:0.8
DEFINE: Sustain_On "ControlChange 64 data:127"

SEQUENCE: Mystic [C3 H F#3 Bb3 E4 A4 D5 C6]
LOAD: Bank0 Prometheus

0:0 Piano Sustain_On
0:0 Piano Play Mystic accent:classic intensity:0.6 humanize:0.5

... 
! Improvise 4 bars using the Prometheus database
24:4 Piano Genplay Bank0 intensity:0.6 transpose:-3

80:0 End
```

Most of the syntax is self-explanatory. There are two sorts of expressions: directives and instructions. Directives are shown in capital characters. As an example, *DEFINE* is a preprocessor directive, inspired by the C programming language, specifying a name and a replacement text. This is useful for creating macros and extending the expressivity of the language. An important structure is *SEQUENCE*, which is basically an array of notes. It uses the classical

letter notation with the special cases of R (rest), H (hold) and X (variable).

The language enables to represent a score by using instructions of the following form:

Bar:Pulse Instrument Instruction parameter:value ...

An example of a high-level instruction is *Genplay* that generates sequences using the corpus-based Genetic Algorithm (cf. previous code example).

The resulting program code can be uploaded in a performance-oriented web-based interface (cf. figure 2).

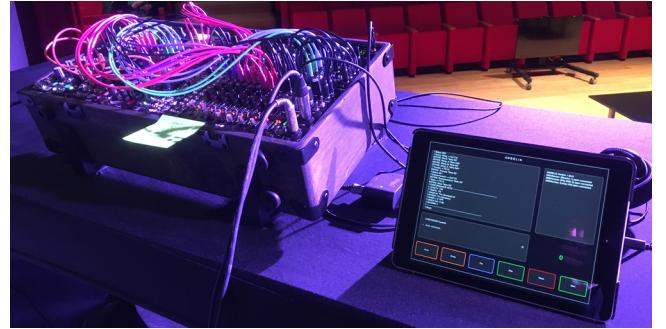


Figure 2. Hardware configuration showing the Modular Synthesizer and the tablet running ANGELIA. The performance-oriented interface includes a window displaying the script (top left); another one displaying the generated Midi flow (top right), and a live coding console (bottom left). In addition, there is a series of assignable buttons for controlling the execution (bottom).

## Bio-inspired Algorithms

ANGELIA is not based on a single algorithmic approach. Instead, its programming language enables to choose for each instruction among different bio-inspired algorithms. It includes generative instructions based on the following kinds of algorithms:

- Procedural and stochastic generators,
- Evolutionary Algorithms,
- Cellular Automata,
- Fractal development,
- Neural Networks.

As an example, ANGELIA includes a dedicated Corpus-based Evolutionary Algorithm. One of the first Evolutionary Algorithms applied to music was *GenJam*, a Genetic Algorithm for generating jazz solos [15]. Like *GenJam*, our algorithm is inspired by natural selection among a population of individuals, the process that drives biological evolution [16]. By using this approach, a sequence of notes represents the “genetic code” of a melody or a chord progression. The population of musical sequences evolves over successive generation by breeding, through selection, crossover and mutation. Each genotype can be then developed to its phenotype, i.e. musical expression, in the envi-

ronment, i.e. the musical piece. The selection of a candidate, for reproduction and expression, depends on its fitness evaluation. The fitness function is a multi-parameter procedural function that scores each individual based on consonance calculation [17] and structural analysis [18].

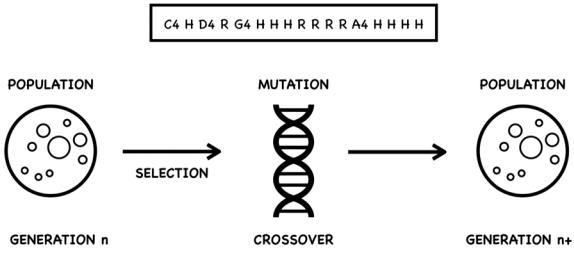


Figure 3. Simplified principle of the Genetic Algorithm. Individuals are 4-bar long sequences in the current implementation.

In contrast with classical genetic algorithms, our implementation does not start from a random generated population, but is initialized using a corpus database. ANGELIA does not use large volume of data from uncited composers, like most Neural Networks approaches. The database includes a carefully curated corpus of patterns from both classical and jazz composers, including Chopin, Litsz, Bach, Debussy, Corea, Jarrett, among others.

## Emotional Metabolism

In most AI music projects, the system generates music with no direct feedback from the produced sounds in the environment. In parallel with the generation of music, ANGELIA analyzes the perceived sound environment in order to generate stimuli that update an “emotional metabolism.” This module in turn influences parameters that modify the expressiveness of the interpretation.

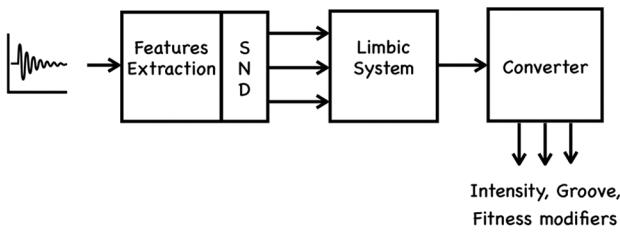


Figure 4. Block diagram of the emotional feedback loop based on three unsupervised neural networks. SND represents the three virtual neurotransmitters: Serotonin, Noradrenaline, and Dopamine.

The emotional metabolism is based on previous works about emotional virtual characters [19] [20]. It is composed of three main unsupervised neural network modules: the analyzer, the “limbic” system and the converter.

The analyzer extracts features from the perceived sound and transforms them into three virtual neurotransmitters: (1) *Serotonin* is an inhibitory stimulus that increases positive vs. negative feelings; (2) *Dopamine* is both excitatory and inhibitory, and related to pleasure and the reward-learning process; (3) *Noradrenaline* is an excitatory stimulus that is responsible for increasing active vs. passive feelings.

The limbic module is based on an emotional model inspired by works on the PAD(Pleasure-Arousal-Dominance) [21] and the Lövheim models [22]. It represents emotions, affects and moods in a three-dimensional space, where the three virtual neurotransmitters form its axes. Therefore, the current emotional state is a moving point in this 3D finite space, where the eight basic emotions, labeled according to the Affect Theory [23] are placed in the eight corners of the cube.

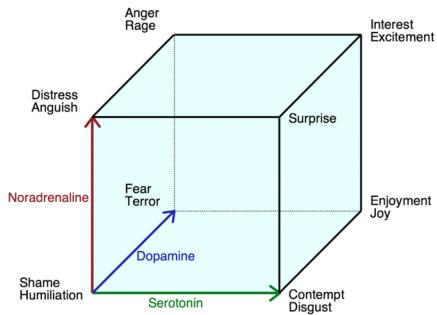


Figure 5. Mapping of the main emotional states on the three-dimensional space. The axes correspond to the three virtual neurotransmitters cumulated values.

The converter takes the coordinates of the current emotional state and converts them into values that updates the expressiveness of the interpretation, such as intensity, velocity, and swing parameters. In addition, it can be also used to modify fitness values when selecting melodic lines in the current population of the corpus-based genetic algorithm.

## Hyperorchestration

ANGELIA can be defined as a Hyperinstrument: a musical instrument capable of playing multiple voices with extended composing and playing capabilities using Artificial Intelligence.

Sergi Casanelles coined the term “hyperorchestra” in his Ph.D. thesis [24]. He defined it as a new approach to the creation of contemporary music for audiovisual media. The term itself is derived from the concept of hyperreality, as defined by Umberto Eco [25], among others. Thus, the term is the portmanteau of “hyperreal” and “orchestra,” which implies a musical ensemble that inhabits hyperreality. While his definition focused on contemporary movie music using sample libraries, we generalize the approach

in order to apply it in the larger context of electronic music creation.

Hyperorchestration expands the classical concepts of orchestra, orchestration and instruments. A classical orchestra can be generally broken down into four main primary groups: strings, woodwinds, brass, and percussion. In contrast, a hyperorchestra is composed of an arbitrary number of groups, each of them having a set of musical instruments, including hyperinstruments. Groups, instruments and listeners are not placed according to the typical orchestra-seating chart resulting in a conventional stereo field, but in a spherical virtual diegetic space. Thus, Hyperorchestration can be defined as the set of approaches, methods and guidelines in order to choose instruments, to place them in the hyperreal space, and to achieve a good cohesion and balance between them.

## Conclusion

In this paper, we have introduced the ANGELIA research project for Electronic Music. We have described its main principles and global architecture in the framework of the Hyperorchestration approach. Forthcoming papers will describe with more details each part.

Future developments includes the integration of new algorithms based on Markov Chains and advanced Neural Network models, but also additional corpus from both classical and jazz composers. Further developments and experiments on the emotional metabolism are also planned.

ANGELIA is not yet another AI project applied to music, but above all it is a Music project using AI. In this framework, we have released albums that retrace the artistic evolution of the project. They can be freely listened or downloaded on an independent and open music platform [26].

## References

- [1] Karl H. Wörner, *Stockhausen: Life and Work*, trans. Bill Hopkins (Berkeley: University of California Press, 1973).
- [2] Iannis Xenakis, *Formalized Music: Thought and Mathematics in Composition*, 2<sup>nd</sup> revised edition (Sheffield: Pendragon Press, 1992).
- [3] James Pritchett, *The Music of John Cage* (Cambridge: Cambridge University Press, 1993).
- [4] Douglass R. Hofstadter, *Gödel, Escher, Bach: an Eternal Golden Braid* (New York: Basic Books, 1979).
- [5] Lawrence M. Zbikowski, *Conceptualizing Music: Cognitive Structure, Theory, and Analysis* (Oxford: Oxford University Press, 2002), 142–143.
- [6] G. Papadopoulos and G. Wiggins, “Ai methods for algorithmic composition: A survey, a critical view and future prospects,” (Edinburgh, 1999). *AISB Symposium on Musical Creativity*, 110–117.
- [7] David Cope, *Virtual Music* (Cambridge: The MIT Press, 2001).
- [8] Jean-Pierre Briot, Gaëtan Hadjeres and François Pachet, *Deep Learning Techniques for Music Generation* (Cham: Springer, 2020).
- [9] Donna Haraway, “A Cyborg Manifesto: Science, Technology, and Socialist-Feminism in the Late Twentieth Century” *Socialist Review* 80, (1985): 65–108.
- [10] Anastasia Georgaki, “The Grain of Xenakis’ Technological Thought in the Computer Music Research of our Days” (Athen, May, 2005). *Proceedings of the International Symposium Iannis Xenakis Proceedings*, 355–361.
- [11] John T. Lysaker, *Brian Eno’s Ambient 1: Music for Airports* (Oxford: Oxford University Press, 2018).
- [12] Ge Wang and Perry R. Cook, “On-the-fly Programming: Using Code as an Expressive Musical Instrument” (New York, 2004). *Proceedings of the 2004 International Conference on New Interfaces for Musical Expression*.
- [13] Chris Wilson and Jussi Kalliokoski, “Web MIDI API”, The World Wide Web Consortium (W3C), accessed May 31, 2022, <https://www.w3.org/TR/webmidi/>
- [14] “Music Tracker”, Wikipedia, the free encyclopedia, accessed May 31, 2022, [https://en.wikipedia.org/wiki/Music\\_tracker](https://en.wikipedia.org/wiki/Music_tracker)
- [15] John A. Biles, “GenJam: A Genetic Algorithm for Generating Jazz Solos” (Aarhus, Michigan Publishing, 1994). *Proceedings of International Computer Music Conference*, 131–137.
- [16] David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Boston: Addison Wesley Publishing Company, 1989).
- [17] Norman D. Cook and Takefumi Hayashi, “The Psychoacoustics of Harmony Perception”, *American scientist* 96, (Sigma Xi, The Scientific Research Society, 2008): 311–319.
- [18] Kjell Bäckman, “Automatic Fitness in Evolutionary Jazz Improvisation” (Belfast, 2008). *Proceedings of the International Computer Music Conference*.
- [19] Jean-Claude Heudin, “A Bio-inspired Emotion Engine in the Living Mona Lisa”, *Proceedings of the Virtual Reality International Conference* (Laval, 2015): 1–4.
- [20] Jean-Claude Heudin, “An Emotional Multi-personality Architecture for Intelligent Conversational Agents”, in *Transactions on Computational Intelligence XXVIII* (Springer, 2018): 1–21.
- [21] Albert Mehrabian, Pleasure-Arousal-Dominance: A General Framework for Describing and Measuring Individual Differences in Temperament”, *Current Psychology* 14(2), (1992): 261–292.
- [22] Hugo Löveheim, “A New Three-Dimensional Model for Emotions and Monoamine Neurotransmitters”, *Med. Hypotheses* 78, (2012): 341–348.
- [23] S. S. Tomkins, *Affect Imagery Consciousness* vol. I–IV (New-York: Springer, 1991).
- [24] Sergi Casanelles, “The Hyperorchestra: A Study of a Virtual Musical Ensemble in Film Music that Transcends Reality”, (Ph.D. diss., Steinhardt School of Culture, New York University, 2015).
- [25] Umberto Eco, *Travels in Hyperreality* (New York: Mariner Books, 1990).
- [26] Jean-Claude Heudin, “Angelia”, accessed June 2, 2022, <https://angelia.bandcamp.com/>

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/339426560>

# Argument harvesting using chatbots

Article in *Frontiers in Artificial Intelligence and Applications* · September 2018

---

CITATIONS

12

4 authors:



Lisa Andreevna Chalaguine

University College London

13 PUBLICATIONS 94 CITATIONS

[SEE PROFILE](#)



Anthony Hunter

University College London

283 PUBLICATIONS 7,752 CITATIONS

[SEE PROFILE](#)

---

READS

84



Fiona L Hamilton

University College London

68 PUBLICATIONS 1,225 CITATIONS

[SEE PROFILE](#)



Henry W W Potts

University College London

280 PUBLICATIONS 7,580 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Outcome Measures [View project](#)



Feasibility RCT: Digital Alcohol Management On Demand (DIAMOND) [View project](#)

# Argument Harvesting Using Chatbots

Lisa A. CHALAGUINE<sup>a</sup> Fiona L. HAMILTON<sup>b</sup> Anthony HUNTER<sup>a</sup>  
Henry W. W. POTTS<sup>c</sup>

<sup>a</sup>Department of Computer Science, University College London, London, UK

<sup>b</sup>eHealth Unit, University College London, London, UK

<sup>c</sup>Institute of Health Informatics, University College London, London UK

**Abstract.** Much research in computational argumentation assumes that arguments can be obtained in some way. Yet, to improve and apply models of argument, we need methods for acquiring them. Current approaches include argument mining from text, hand coding of arguments by researchers, or generating arguments from knowledge bases. In this paper, we propose a new approach, which we call *argument harvesting*, that uses a chatbot to enter into a dialogue with a participant to get arguments and counterarguments from him or her. Because it is automated, the chatbot can be used repeatedly in many dialogues, and thereby it can generate a large corpus. We describe the architecture of the chatbot, provide methods for clustering arguments by their similarity and value, and an evaluation of our approach in a case study concerning attitudes of women to participation in sport.

**Keywords.** argument harvesting, chatbots, value-based argumentation, behaviour change

## 1. Introduction

Abstract argument graphs, such as proposed by Dung [10], are an important formalism in computational models of argument. However, the issue of acquiring the graphs tends to be omitted. In order to construct graphs using *real* arguments as opposed to theoretical, made-up scenarios, arguments have to be acquired from real-life sources. A common approach to argument acquisition assumes a static resource available on the internet where the topic of interest is/was already discussed. This, however, raises several problems: firstly, what if no discussion platform for a particular topic exists? Secondly, even if it exists, what if not enough representative people contribute to the opinion exchange? Thirdly, such platforms do not take into account attributes of the individuals who posited the arguments. This is a drawback of other systems such as D-BAS [11] that are more suitable for public argumentation or collective decision making where *all* existing arguments on a particular topic are of interest. We, on the other hand, focus on behaviour change which requires a more individual approach. The only solution is then to use questionnaires or to interview people directly. That, however, may be a labour-intensive and expensive undertaking. To address these issues, we believe that it is possible to automate the process of argument acquisition using chatbots. A chatbot is a computer program that can chat with humans via text. As a proof of concept, in this paper, we present a method focused on argument acquisition for behaviour change applications but which could be adopted to other application domains as well.

Recently chatbots have been developed for domains like health care and behaviour change [2,3]. Human agents respond and converse with artificial agents in ways that to some extent mirror emotional and social discourse dynamics when discussing behavioural health [9]. Therefore, there is literature to suggest that using a chatbot to acquire user arguments on a certain behaviour and address the problems of traditional argument acquisition, is possible. As already shown by Weizenbaum, a chatbot that uses only generic questions is indeed capable of encouraging the user to talk about himself [15]. His chatbot *Eliza* simulated conversation by using pattern matching and pronoun substitution that gave users an illusion of understanding even though it had no built-in knowledge. This therefore indicates that generic questions may enable a chatbot to harvest arguments in diverse domains.

So far, no attempts of using a chatbot for argument acquisition have been made in the computational argumentation domain. In this paper, we investigate the approach at *argument harvesting* which we define as acquiring arguments with the help of a chatbot. We further perform three experiments with crowd-sourced participants in order to analyse the meaningfulness, values and relationships of the arguments. The contribution of our work is threefold: first, we describe a model for argument harvesting using a simple chatbot with little or no domain knowledge. Second, we show that people who give the same or a semantically similar argument, are most likely motivated by the same value when positing it. We demonstrate that it is therefore possible to train a classifier to predict the value (motivation) of an argument. And third, we present a method to cluster the harvested arguments by value and semantic similarity in order to automatically create several possibilities to counter a given argument.

The rest of the paper is structured as follows: Section 2 gives some background theory on value-based argumentation frameworks, an overview on values, as well as our own definitions of values; Section 3 gives the aim of the paper and the hypotheses; Section 4 describes the chatbot architecture that was used for argument harvesting; Section 5 describes the experiments that were conducted throughout the study including their methodology and results, and in Section 6 we discuss and conclude our findings.

## 2. Values in Argumentation

In order to account for different points of view in debates, it has been recognised that the parties within a debate will have different perspectives on what is important to pursue, according to their subjective aspirations and preferences [5]. In value-based argumentation [7,8] arguments promote specific values which account for the social interests of debate participants. Values are assigned to an argument when constructing argument graphs. They provide an explanation as to why it is not always possible to persuade others to accept an opinion simply by demonstrating facts and proofs. It may be that a particular individual will accept the facts of a decision but will reject the conclusion to act upon it because it does not support the values he or she holds [5]. Although we do not use value-based argumentation frameworks in this paper, we are interested in the notion of values and their relevance to argumentation in behaviour change.

None of the papers that apply value-based argumentation frameworks to specific examples [4,14,13] explain where the values come from or according to what principles they should be chosen. We therefore need to define our own notion of values for our

purpose. In a dialogue when someone posits an argument, they normally have some *motivation* for choosing *this* specific argument as opposed to another in that part of the dialogue. We call these *categories of motivation* which are categories that are important to the *life* of the agent. Note, we are not interested in motivations concerning the dialogue, e.g. winning the dialogue, revenge, showing-off, deceiving etc.

In this paper we study attitudes of women towards engaging in physical exercise. We are concerned with the notions of *value* of an argument and the *suitability* of a counterargument. The following example illustrates the two notions: given the categories of motivation for not exercising  $V = \{\text{family, comfort, dignity, wealth}\}$ , suppose a woman (the persuader) is trying to convince her friend (the persuadee) to do more sports and gives the following argument: **A1:** “*Physical activity is healthy and you should therefore go to the gym more often.*” The persuadee, assuming she is rational, will not try to counter the fact that physical activity is healthy and will most likely *accept* that fact. She may, however, counter the conclusion (which action to take) with an argument that reflects her motivations (values) for not engaging in physical activity. She might say: **A2:** “*I have no time because I have to look after my kids.*” In this case the argument promotes the value *family*. To generalise this idea, we give the following definition for values which delineates how we can assign a value to an argument.

**Definition 1** A **value assignment** by an agent to an argument  $A$  is a category of motivation for the agent if the agent were to posit  $A$ .

In the above definition we use the phrase “if the agent were to posit  $A$ ” because we will investigate how individuals value arguments independently of a specific dialogue.

In this paper, we are interested in a certain *kind* of counterargument that is appropriate for dialogues in behaviour change. For such dialogues, we believe a counterargument should have the same value assignment as the argument it attacks. Continuing with the example above, the persuader would respect the value *family* and give a counterargument A3 that attacks A2 but respects the value *family*. For example: **A3:** “*You could incorporate your children into your exercise routine. Like going roller blading in the park or swimming.*” So A3 attacks A2 while respecting the same value and still pursues the initial intention of persuading the persuadee to do more sports. This does not mean that the persuadee has to agree with the given counterargument, it merely means that the counterargument can be given as a *suitable* counterargument to the previously posited argument. We define the notion of *suitability* of a counterargument next:

**Definition 2** Let  $A$  be an argument and let  $CA$  be a counterargument that attacks  $A$ .  $CA$  is a **suitable** counterargument to  $A$  iff  $A$  has a value assignment  $V$  and  $CA$  has a value assignment  $V'$  such that  $V = V'$ .

In this section we have outlined the importance of values in argumentation for behaviour change and have given our own definition of *value assignment* to arguments. We have also introduced the concept of *suitability* for counterarguments that can be used to counter a previously posited argument that promotes a specific value. Given these notions of value and suitability, we want to test several hypotheses, given in the next section.

### 3. Hypotheses

In this paper, we make a first step towards argument harvesting. We chose *attitudes of women to participation in sport* as a case study. We have developed a chatbot that harvests arguments and their values from women on why they do not engage in (more) physical activity. The chatbot also asks them to provide suitable counterarguments to their given arguments (more on the dialogue protocol in the next section). Each argument therefore has a value and a counterargument. Given this, we want to test three issues: first, whether different people are motivated by the same value if giving the same, or semantically similar argument. Second, whether our chatbot is capable of harvesting meaningful arguments i.e. those considered to be appropriate arguments by sufficiently many participants from the people group the argument was harvested from. Third, whether we can automatically match an argument with more suitable counterargument and therefore create more possibilities to counter a certain argument. We summarise these points in the following three hypotheses:

- H1** The majority of people that exposed to, but not necessarily posit, the same argument, assign to it the same or similar value, therefore making it possible to predict the value of an argument.
- H2** A domain neutral chatbot, with little or no domain specific knowledge, and by giving general responses, can acquire arguments that are perceived as meaningful by the people group the arguments were harvested from.
- H3** Given arguments semantically similar in meaning with the same value, counterarguments are interchangeable making it possible to use the counterargument of one argument as a counterargument to another argument.

In the remainder of this paper we describe the design of our chatbot that was used for argument harvesting and explain the experiments conducted with the harvested arguments in order to test our hypotheses.

### 4. Chatbot Design for Argument Harvesting

Messaging has become the most widely used communication layer on mobile platforms during the last few years, with Facebook Messenger (FM) being the most popular messaging application<sup>1</sup>. FM is a free instant messaging service and software application which lets Facebook users chat with other users (or chatbots) on the main website as well as the mobile app. For building chatbots, the Messenger Send API gives the ability to send and receive messages. Due to the popularity of FM and the free API that Facebook provides we decided to use FM as the platform to deploy our chatbot.

We created an application called *ArgHealthBot* which users can send messages to. The application is linked to a Facebook page which has a *Send Message* button. The page also displays a link to a website that contains the terms and conditions of the chatbot and states that we received ethical approval for our study, and a short description of the current experiment. For the screenshots of the website and the application, see Appendices G and H [1]. When users click on *Send Message*, a FM window pops up and allows them

---

<sup>1</sup>1.3 billion active users as of December 2017

to send private messages to the application to which the chatbot is connected. The chatbot code is written in the Python programming language and consists of a Flask server and the text-processing code. The server code communicates with the Send API and the text-processing code processes the incoming messages from users and sends appropriate responses.

The dialogue protocol is the following: after the participant initiates the chat and consents to continue with the experiment, the chatbot asks to provide a reason for why she is not engaging in (more) physical activity, to which the participant answers with an argument (A1). If the chatbot considers the answer too short (less than 12 words) it asks to expand on the given argument. The chatbot queries the participant to expand on the argument only once. The expansion of the argument (if there is one) is added to the initial argument and the complete, harvested argument is added to the argument database. The pseudo-code and a description of the algorithm for query-generation (asking to expand on a given answer) can be found in Appendix I [1].

In order to assign values to the arguments we needed a set of values to choose from. We used the list of personal values from Scott Jeffrey<sup>2</sup> as reference and pragmatically chose values that we found suitable. The values were: *responsibility, comfort, dignity, satisfaction, relaxation, family, friendship, professionalism, productivity, wealth, knowledge, fun, recreation, ambition and safety*. The chatbot presents the user with the list of values after she provided an argument and asks to choose the one she most associates with her argument.

The chatbot then asks what the user would recommend a friend with the same problem. This is the counterargument to the previously given argument (CA1). The chatbot picks up on that and asks why the user is not following her own advice. The user answers with another argument (A2). The chatbot asks again what she would advise a friend with the same problem (CA2). After harvesting two argument-counterargument pairs the chatbot asks the participant whether she wants to continue or end the chat. Our chatbot therefore harvests a minimum of two argument-counterargument pairs {(A1, CA1), (A2, CA2)}.

## 5. Experiments

In this section, we describe how we collected the arguments concerning women's participation in sports via argument harvesting (AH) and the experiments conducted with the harvested arguments. For each experiment we give the purpose, the methods used, the results and conclusion of our findings. The participants for all experiments were recruited via *Prolific*<sup>3</sup>, which is an online recruiting platform for scientific research studies. For each experiment we recruited from three disjoint groups: students (aged 18-25 and no children), women with children (aged 18-40 and not students) and women without children (aged 18-40 and not students), in the following referred to as the *student*, *kids* and *nokids* groups respectively. We opted for this division in order to get a wider spectrum of different arguments from different people groups, or *audiences*. For each experiment we evaluate how the arguments are perceived by the audience it is meant for, based on the assumption that a particular argument is addressed to a specific audience [6]. The

---

<sup>2</sup><https://scottjeffrey.com/core-values-list/>

<sup>3</sup><https://www.prolific.ac/>

general prerequisites for taking part in our study were to be female, over 18 and engaged in less than 150 minutes of physical exercise per week. For the argument harvesting, we required the participants to have a Facebook account in order to chat with the chatbot. For the experiments, Google Forms were used.

### 5.1. Argument Harvesting

We conducted two rounds of argument harvesting (referred to as AH1 and AH2). In AH1, we used our chatbot to harvest arguments and their associated values and counterarguments from the three participant groups. In AH2, we harvested arguments and counterarguments without their values.

For AH1, we recruited 30 participants for the *student* group, 30 for the *kids* group and 50 for the *nokids* group. The women who participated in the study and agreed to chat with the chatbot, initiated the conversations and the chat followed the dialogue protocol described in the previous section. For an example of a chat between participant and chatbot see Appendix E.1 [1].

Dialogues where participants described certain medical conditions like social anxiety, depression and scoliosis were removed from the data (10 dialogues in total). We decided that those require professional consultation and should not be included in this study. We also narrowed down the set of values by disregarding values that appeared in the whole data less than 5 times. The dialogues where at least one of the arguments had a deleted value, were removed (18 dialogues in total). The values used for the following experiments were: *responsibility, family, productivity, dignity, wealth, comfort, relaxation and fun*.

For AH2, 20 participants from each group were recruited and asked to chat with the chatbot. This time we included more prerequisites during the recruitment, namely no chronic diseases, no long-term health conditions/disabilities and no ongoing mental illnesses. In this round, the chatbot did not ask the participants to assign values to their arguments. For an example of a chat, see Appendix E.2 [1]. We harvested 40 arguments for each participant group in AH2 (no dialogues were deleted). The total number of argument-counterargument pairs after the two rounds of argument harvesting was 284 and can be found in Appendix A [1].

After AH1, we made the following three observations. Firstly, some values were chosen more often than others and a smaller set of values therefore suffices to cover most of the arguments. Secondly, our simple chatbot was capable of harvesting a significant number of arguments. And lastly, we observed that many participants gave similar arguments or even the same argument, using different words. This opens the possibility of grouping arguments using clustering techniques. The experiments we conducted with the harvested arguments in order to test our hypotheses, are described in the following.

### 5.2. Experiment I: Argument-Value Labeling

The purpose of the experiment was to test whether different people assign the same (or similar) values to the same arguments that they have not posited themselves and whether it is possible to *predict* the values of arguments by training a classifier and therefore verify Hypothesis I.

The methods used in the experiment were the following. 20 participants for each group were recruited using the same prerequisites as for the argument harvesting, apart

**Table 1.** Average agreement (AGT) for values (V) and parent-values (PV) for arguments harvested in AH2 and the corresponding kappa scores ( $\kappa$ ).

Group	S	K	NK
V AGT	68.31%	62.56%	66.86%
V $\kappa$	0.40	0.27	0.40
PV AGT	81.45%	86%	81.43%
PV $\kappa$	0.52	0.42	0.49

**Table 2.** Accuracy (AC) of the classifier-predicted values (V) and parent-values (PV) and the corresponding  $F1$  scores.

Group	V AC	V F1	PV AC	PV F1
S	50%	0.47	77.5%	0.77
K	55%	0.55	82.5%	0.84
NK	42.5%	0.45	70%	0.69
Avg	49.2%	0.49	76.7%	0.77

from the Facebook account, as no chatting with the chatbot was required. We used Google Forms for this task. Since we were interested in how the same group of people judged the arguments, we asked members of the *student* group to assign values to the arguments given by the students (respectively for the *kids* and *nokids* groups). The participants were presented the 40 arguments from their group harvested in AH2 and given a choice of 8 values. They were asked to “read the argument for not engaging in physical activity and pick the value that they associated with the given argument”. The value that received the highest vote amongst the participant (value agreement) was chosen as the corresponding value for that particular argument. For example, if for argument A1, 16 out of 20 participants chose the value *family*, then *family* was assigned to A1 and the value agreement is 80%.

We observed that certain values are interchangeable: for example, the value ‘responsibility’ was equivalent to ‘family’ in the *kids* group and ‘productivity’ in the *student* group. We therefore grouped six out of the eight values into the following two groups, calling these *parent-values*: **CRF**: *comfort, relaxation and fun*. **FRP**: *family, productivity* and *responsibility*. The remaining two values *wealth* and *dignity* had no parent-value<sup>4</sup>. Parent-value agreement for the individual arguments was calculated by adding up the agreement rates for the individual values in that parent-value group. The agreement ratios for the individual groups (abbreviated *S*, *K*, *NK* for the *student*, *kids* and *nokids* groups respectively) are shown in Table 1. We also calculated *Fleiss Kappa* scores in order to assess the reliability of agreement between the participants of each group. For the value agreements for individual arguments, see Appendix B [1].

We used the values assigned by the participants that received the highest value agreement (participant values) to score the value-classifier. The arguments and values from AH1 were used for training, while the arguments from AH2 and the participant values were used for testing. We trained a Support Vector Machine with a linear kernel using the bag-of-words model. We scored the classifier by comparing the classifier-predicted values to the participant values. The results are shown in Table 2. Accuracy is defined as the number of arguments where the value predicted by the classifier was the same as the participant value. There was a choice of 8 values and 3 parent-values. Random classification would therefore be 12.5% and 33.33% respectively. Our classifier had an average accuracy of 49.9% for the values and 76.7% for the parent-values. Table 2 also includes the weighted  $F1$  scores for each participant group.

<sup>4</sup>They were grouped together as a parent-value during the classification in order to create a bigger group for the classifier as the two values on their own had too few examples.

**Table 3.** Distribution of arguments (Args) with parent-values *FRP*, *CRF*, and values *Dignity* and *Wealth* (the classifier-predicted values are used for the arguments from AH2).

Group	No. of Args	FRP	CRF	Dignity	Wealth
S	80	31.25%	60%	1.25%	7.5%
K	92	72.83%	25%	0%	1.09%
NK	112	25.89%	67.86%	1.79%	3.57%

The accuracy of prediction for the *nokids* group is lower than the other two groups due to the more diverse arguments compared to the *kids* group. Table 3 shows how many arguments in each group are assigned with a specific parent-value. In the *kids* group, 72.83% of the arguments are assigned the values *family* or *responsibility*. These arguments often contain the words *children*, *baby* and *kids*. For the *nokids* group the majority of the arguments (67.86%) have the values *comfort*, *relaxation* and *fun*. Those arguments are much more diverse and do not have as many keywords in common which makes classification more difficult.

It can be concluded that even though people might disagree on nuances like whether a certain argument promotes the value *family* or *responsibility* in the *kids* dataset or cannot decide whether an argument given by a person is better associated with *relaxation* or *comfort*, the majority of people agree on the parent-value for a given argument. On average, participants agreed 65.9% on the value and 83% agreed on the parent-value. The results therefore support our Hypothesis I, that the majority of people independently assign the same or similar values to an argument that they have not posited themselves.

### 5.3. Experiment II: Assessment of Harvested Arguments as Meaningful Arguments

In this experiment, we wanted to assess whether a chatbot can be used as a tool for harvesting meaningful arguments and therefore verify Hypothesis II.

The methods used in the experiment were the following. We recruited 10 participants for each group (like in the previous experiment, participants were representatives of the groups, e.g. students judging the arguments given by students). The prerequisites were the same as in Experiment I. Participants were presented all 40 arguments harvested of the corresponding group in AH2 in a Google Form. We told the participants that the arguments were crowd-sourced reasons for not exercising and asked them whether they “considered the given arguments as reasons they could give an appropriate advice”. We also asked them to not judge the quality of the reason, rather just the completeness of it. After each argument they had the choice of selecting *yes* or *no*.

The results of the experiment are summarised in Table 4. We explain how we derived the results as follows: We set the threshold for considering a statement as an argument at 70% annotator agreement (approval rate). This means that if a minimum of 7 out of the 10 participants answered the question whether a given statement is an argument positively, we labeled it as *meaningful*. For the results for the individual arguments, see Appendix C [1].

From the results, it can be concluded that a chatbot can indeed harvest meaningful arguments using no or very little domain knowledge, which supports our Hypothesis II. In total over 78% of the arguments that were harvested in AH2 were considered meaningful.

**Table 4.** Meaningful arguments (Args) in each group when the threshold is set to 70% annotator agreement and above

Group	No. of Args	No. of meaningful Args
S	40	28 (70%)
K	40	33 (82.5%)
NK	40	33 (82.5%)

#### 5.4. Experiment III: Argument-Counterargument Matching

The purpose of the experiment was to test Hypothesis III i.e. to evaluate whether the counterarguments of semantically similar arguments are interchangeable, making it possible to use the counterargument of one argument to counter another similar argument.

The methods used in the experiment were the following. In order to cluster similar arguments we needed a clustering algorithm. Our dataset was too small to apply general-purpose unsupervised clustering algorithms, so we developed a specialised clustering algorithm that could take advantage of domain specific knowledge. We describe the algorithm below and the pseudo-code can be found in Appendix J [1].

First, we create a synonym list using WordNet [12]. This list contains lists of all the words in a given corpus that are synonyms of each other. Then the arguments are normalised by deleting stopwords and punctuation, and setting the case to low. We also delete exercise and time related words (*exercise/s, sport/s, day/s, week/s, hour/s, thing/s, reason/s, main, lot*) because a lot of people repeated the chatbot’s question in their answer (e.g. “*The main reason I don’t exercise is [...]*”). So we did not want to consider those in our similarity measurements. We also disregarded words that were used to describe how often they did or did not engage in a certain activity. Finally, for each argument, the noun phrases are extracted and stored as separate words and the synonyms are replaced with the first word in the corresponding synonym list. The arguments are stemmed in order to avoid treating different forms of a word as different words. After preprocessing the arguments, all arguments with the same value are clustered by comparing them to each other and clustering those together that share more than 50% of the words. This results in clusters where each argument shares over 50% of words with every other argument. An argument can occur in more than one cluster.

We applied the algorithm separately on the arguments of each participant group (see Appendix F for the clusters [1]). Every argument has an original counterargument as given by the same participant during the chat with the chatbot. Each argument that appeared in a cluster (was ‘clustered’) was matched with all the counterarguments from the other arguments in that cluster, apart from its original one. For example, if the arguments A1, A2, and A3 formed a cluster, then A1 would be matched with counterarguments of the other two arguments CA2 and CA3.

We evaluated the suitability of the counterarguments as follows: 10 participants for each group were recruited, with the same prerequisites as in Experiments I and II. We again used a Google Form where each argument was presented with its matched counterarguments and the participants were asked to choose which ones they believed was a suitable counterargument for the argument given. They were told that the arguments as well as the counterarguments were collected via crowd-sourcing and that they should not judge the quality of the arguments and counterarguments, but rather whether the counterargument is an appropriate response to the given argument.

**Table 5.** Total number of arguments (Args) in each group, number (percentage) of arguments clustered, the average number of counterarguments (CAs) per clustered argument and the number of argument clusters generated in each group.

Group	No. of Args	Clustered total (%)	Clustered AH1 (%)	Clustered AH2 (%)	Avg CAs	Clusters
S	80	40 (50%)	18 (45%)	22 (55%)	3.65	19
K	92	49 (53.26%)	23 (44%)	26 (65%)	7.39	22
NK	112	42 (37.5%)	24 (33%)	16 (40%)	6.62	14

The results of the experiment are summarised in Tables 5-7. Table 5 shows how many arguments were clustered in the individual groups and the two rounds of harvesting. We can see that in the *nokids* group fewer arguments were clustered than in the other two groups. This is due to the higher diversity in arguments and more complex synonyms.

The counterarguments of each argument received a certain approval rate, showing how often a given counterargument was selected by a participant. Table 6 (column 3) shows the average approval rates of the counterarguments for each argument in that group. For example if an argument had three counterarguments and the approval rates of them were 20%, 70% and 90%, the average approval rate of the counterarguments for that argument would be 60%. For more examples see Appendix D [1].

We considered the average number of suitable counterarguments per argument by using an approval rate threshold of 50%. If, for instance, an argument had three counterarguments with the approval rates 40%, 50% and 60% respectively, the second and third would be considered suitable and the number of suitable counterarguments would be 66.7% (2/3). The results are shown in Table 6 (column 4). The reason for the lower threshold is the high variance of quality amongst counterarguments. Some counterarguments scored poorly because they give inappropriate advice (see Example 1).

**Table 6.** Average approval rate (AR) of counterarguments (CAs) per argument and the average number of suitable CAs per argument with approval threshold of 50%.

Group	No. of Args	Avg. CA AR	Avg. No. suitable CAs
S	40	70.37%	80.66%
K	49	69.04%	84.41%
NK	42	60.10%	78.89%

**Table 7.** The average approval rate (AR) of individual counterarguments (CAs) when matched with the corresponding arguments in their cluster.

Group	No. of CAs	Avg. AR
S	40	69.18%
K	46 <sup>5</sup>	72.01%
NK	42	58.82%

We also analysed the approval rate that the individual counterarguments received, averaging all the approval rates that a counterargument received for all the arguments it was matched with. This way we wanted to identify inadequate counterarguments and wrongly clustered arguments. For example, if counterargument CA4 was matched with three arguments A1, A2, A3 and received an approval rate of 40% for A1, 50% for A2 and 80% for A3, the average approval rate for CA4 would be 56.7%. The results are shown in Table 7. The following is an example of an inappropriate counterargument:

<sup>5</sup>There are only 46 counterarguments for the 49 clustered arguments because in three cases the participants answered “I don’t know” instead of giving a counterargument.

**Example 1** The argument A4 and counterargument CA4 were given by the same participant.

A4: “*I only sometimes do sports because I am too busy and tired from my uni work*”.

CA4: “*You could join a sport team with a friend or find a gym buddy*”.

A4 was clustered with similar arguments (a total of 6) and therefore CA4 was matched with all the arguments of that cluster. It was, however, never approved as a suitable counterargument and had the lowest average approval rate in the *student* dataset (17.5%). It is not surprising that this counterargument was not considered a good one. It does not advise on how to manage your time better and/or emphasise the benefits of physical exercise. In the chat, when the chatbot asked why the person was not following her own advice, the participant indeed answered: “*like I said, I am often too busy to do so. I mostly study or try to catch up on sleep*”. A counterargument that can be countered with “*like I said...*” is unlikely to be an appropriate counterargument.

From the results in this section, it can be seen that counterarguments of similar arguments are interchangeable as long as they give appropriate advice, which supports our Hypothesis III about the interchangeability of counterarguments of semantically similar arguments. With the current data participants perceive a counterargument from a similar argument as suitable about 80% of the time, when we set the threshold for suitability at 50% approval rate. Regarding the clustering algorithm, only 131 out of the 284 arguments were clustered. This was due to several factors including wrong classification by the value-classifier, more complex synonyms and lost negations during the preprocessing of the arguments, specific explanations for a common reason, implicit meanings and specific arguments that did not repeat within the data. In the next section we discuss the results of our experiments.

## 6. Discussion

Our contribution in this paper is threefold. Firstly, we have shown that a simple chatbot with little or no domain knowledge can acquire meaningful arguments. We have focused on the behaviour change domain, where ordinary people give simple arguments that are nevertheless full of meaning and importance. They are the kind of arguments that have been neglected in the formal as well as informal argumentation literature. There is little literature on how to analyse this sort of argument and even less on how to acquire them.

Secondly, we have shown that the majority of people assign the same or a similar value to given arguments which makes it possible to predict values of arguments with the help of a classifier. Given this observation, it can be concluded that given an argument, most people will be motivated by the same value if positing it. We also made a first attempt of finding a suitable set of categories of motivation (values) for a specific topic, by letting the participants assign the values to their arguments themselves.

Thirdly, we presented a method to cluster arguments by values and similarity in order to create several possibilities to counter a given argument and evaluated whether the counterarguments of those are interchangeable. The results show that this is the case, given the counterargument itself is appropriate. However, in the future we want to research other methods of counterargument acquisition. One possibility is to harvest the

arguments from one group (e.g. people who do not do sports) and acquire the counterargument from the group with the opposite behaviour (people who do sports). We also want to look into different *types* of counterarguments. In the current study most people, when being asked to give a counterargument, gave *suggestions*. However, there are other ways to counter an argument, e.g. by naming a negative consequence.

Argument harvesting can potentially be used in other domains such as politics, culture and marketing. The advantage over questionnaires is that a chatbot can ask relevant follow up questions and queries with the help of natural language processing. It can therefore acquire more arguments and information on the individual, than a questionnaire on a new political decision, a theater play or a new product could account for.

We want to use the harvested arguments to construct argument graphs and analyse the discourse dynamic in argumentation concerning behaviour change. Our future aim is to develop a chatbot for persuading people to change their behaviour or belief by answering with suitable counterarguments.

## Acknowledgements

We would like to thank Trevor Bench-Capon for his helpful insights on values in value-based argumentation. The first author is funded by a PhD studentship from the EPSRC.

## References

- [1] Appendices. <https://tinyurl.com/y8fjsab8>.
- [2] The future of bots, will they ever be prescribed for healing? <https://tinyurl.com/yb532prq>.
- [3] How chatbots will shape the future of health care. <https://tinyurl.com/ybqflvrv>.
- [4] K. Atkinson. Value-based argumentation for democratic decision support. In *Proceedings of COMMA'06*, pages 47–58, 2006.
- [5] K. Atkinson and A. Wyner. The value of values in computational argumentation. K. Atkinson, H. Prakken, A. Wyner (Eds.), *From Knowledge Representation to Argumentation in AI, Law and Policy Making: A Festschrift in Honour of Trevor Bench-Capon on the Occasion of His 60th Birthday*, College Publications, pages 39–62, 2013.
- [6] T. Bench-Capon. Agreeing to differ: Modelling persuasive dialogue between parties without a consensus about values. *Informal Logic*, 22:231–246, 2002.
- [7] T. Bench-Capon. Value-based argumentation frameworks. In *Proceedings of NMR'02*, pages 443–454, 2002.
- [8] T. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13:429–448, 2003.
- [9] T. Bickmore, A. Gruber, and R. Picard. Establishing the computer-patient working alliance in automated health behavior change interventions. *Patient Education Counseling*, 59(1):21–30, 2005.
- [10] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.
- [11] T. Krauthoff, M. Baurmann, G. Betz, and M. Mauve. Dialog-based online argumentation. In *Proceedings of COMMA'16*, pages 33–40, 2016.
- [12] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- [13] S. Modgil. Value based argumentation in hierarchical argumentation frameworks. In *Proceedings of COMMA'06*, pages 297–308, 2006.
- [14] F. Nawwab, T. Bench-Capon, and P. Dunne. A methodology for action-selection using value-based argumentation. In *Proceedings of COMMA'08*, 172(1):264–275, 2008.
- [15] J. Weizenbaum. Eliza - a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/353325752>

# Chatbot as support to decision-making in the context of natural resource management

Conference Paper · July 2021

DOI: 10.5753/wcama.2021.15734

---

CITATIONS  
0

READS  
50

3 authors:



Bruno Alves  
Universidade Federal de Pelotas

5 PUBLICATIONS 1 CITATION

[SEE PROFILE](#)



Larissa Freitas  
Universidade Federal de Pelotas

35 PUBLICATIONS 189 CITATIONS

[SEE PROFILE](#)



Marilton Sanchotene Aguiar  
Universidade Federal de Pelotas

128 PUBLICATIONS 444 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Comparação Automática entre Textos Escritos por Diferentes Veículos de Comunicação [View project](#)

# Chatbot as support to decision-making in the context of natural resource management\*

Bruno C. Alves<sup>2</sup>, Larissa A. de Freitas<sup>2</sup> and Marilton S. de Aguiar<sup>1,2</sup>

<sup>1</sup>Graduate Program in Computer Science

<sup>2</sup>Technological Development Center

Federal University of Pelotas – Pelotas – RS – Brazil

{bcalves, larissa, marilton}@inf.ufpel.edu.br

**Abstract.** *The management of natural resources is becoming increasingly relevant due to its direct implication in society's life. Thus, individuals must make decisions based on environmental and social aspects. This work uses a chatbot to support users' decisions through an RPG scenario based on the participatory management of resources in the Lagoa Mirim Watershed and Canal São Gonçalo Basin. In this context, in addition to the chatbot, this study presents a pollution predictor to support decision-making, with a determination coefficient of 0.99, constructed using random forest. Also, we present five Word Embeddings models to expand the natural language understanding, based on a corpus of about 700 thousand sentences, capable of identifying relations between words.*

## 1. Introduction

Natural resource management is an area that seeks better ways to manage land, water, plants, and animals, based on the quality of life in society. This area has gained visibility for governments due to sustainable development, which is a principle of how they see and understand the world. Natural resource management has specific objectives the scientific study of resources and how these resources can support life [Holzman 2009]. Water is one of the most important natural resources, as it is essential for social and economic activities [Ponte et al. 2016]. The management of water resources involves different groups and organizations, which need to analyze better ways of distributing and using water.

Considering that this resource is shared and limited, decision-making is a relevant aspect for this management because it is possible to obtain more appropriate solutions through the interaction between individuals [Adamatti 2007]. In this context, Machine Learning (ML) represents systems from computational tools to support risk prediction. Considering the growing presence of chatbots in everyday life [Raj 2019], this type of communication, based on natural language, presents as an alternative for the information propagation that helps in the implementation of actions. Thus, extracting meaning from messages sent to the chatbot can be applied to Natural Language Processing (NLP) and constructed vector representation models with Word Embedding (WE).

The development of this work is in the context of an in-progress research project. In this project, a computational game based on Multiagent Systems (MAS)

---

\*This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES/Brasil) and Agência Nacional de Águas (ANA/Brasil) – Edital 16/2017.

and Role-Playing Game (RPG) for the natural resource management is being implemented, more specifically for the participatory management of water resources in the Lagoa Mirim Watershed and Canal São Gonçalo, located in the south of Brazil. In the RPG [Leitzke et al. 2019], called GORIM, players interpret characters within a story constructed through rules, modeled after interactions with the region's hydrographic basin committee, where players make decisions and communicate with other characters (agents) searching for their individual/collective goals.

Regarding this context, this work presents the development of a chatbot capable of assisting different RPG roles in decision-making. For example, considering environmental information and interactions between characters in a watershed scenario, game agents can consult trends using statistics and make predictions about pollution levels based on a model constructed with ML, applied to the data collected in the RPG pilot sessions. Also, we developed five WEs models to expand the chatbot understanding through the vector representation of words since the resources available in the literature for application in NLP tasks in the Portuguese language are limited.

We organized this article as follows. In Section 2, we present the theoretical background for this work; in Section 3, we describe the technical/methodological decisions that guided the development of this study; in Section 4, we discuss the results obtained; and finally, in the Section 5, we present the conclusions of this work.

## 2. Theoretical Background

This Section will present concepts about Chatbots, ML, NLP, and WE areas in the context of this study. Besides, this Section will discuss the main related works.

A chatbot (also referred to as a conversational agent) is an automated program that seeks to answer questions based on the simulation of human behavior [Raj 2019]. Researchers developed chatbots of various technologies for different purposes in areas such as commerce, school, and health from this event. Currently, there are specific platforms for structuring conversational agents including Watson Assistant<sup>1</sup>, Wit.ai<sup>2</sup>, and Dialogflow<sup>3</sup>. All of these applications use NLP, so it becomes possible to implement and integrate chatbots. According to the complexity of the algorithms used in their construction, we can classify conversational agents based on rules or self-learning. In the rules-based method, the chatbot seeks to answer questions asked according to a set of simple specifications. In contrast, in the self-learning strategy, machine learning techniques are used during conversational agent training [Hussain et al. 2019].

ML can be applied for automatic data analysis to obtain helpful knowledge that assists in resource management and decision-making. In particular, predictive models, based on previous experiences (supervised learning), can be constructed from regression models, which seek to extract patterns from the data and thus predict continuous values [Alpaydin 2014]. Therefore, to understand this work, four regression algorithms will be presented: i) linear regression, this algorithm is an equation that describes the relationship between a dependent variable and a set of attributes; ii) support vector regression

---

<sup>1</sup><https://www.ibm.com/cloud/watson-assistant>

<sup>2</sup><https://wit.ai/>

<sup>3</sup><https://dialogflow.cloud.google.com/>

(SVR), we use this algorithm for seeking a maximum margin that separates the hyperplane to gather the most significant number of data in a narrow area; iii) regression tree, this algorithm is a set of rules based on predictive attributes; and, iv) random forest regression, this algorithm calculates the average of the predictions of a group of regression trees.

NLP is an area composed of techniques that seek to extract meaning from the human's natural language, such as English and Portuguese. Usually, developers apply NLP techniques in chatbots proposed to solve tasks involving the self-learning method, to simplify and standardize the raw text [Eisenstein 2019]. Thus, the main NLP techniques involved are: i) normalization – this technique adequacy the text in terms of spelling, removal of accents, and removal of special characters; ii) tokenization – this technique separates the text in individual terms called tokens; iii) removal of stopwords – this technique removes the words with little relevance; and, iv) lemmatization – this technique transforms the verbs to the infinitive and adjectives/nouns to the masculine singular.

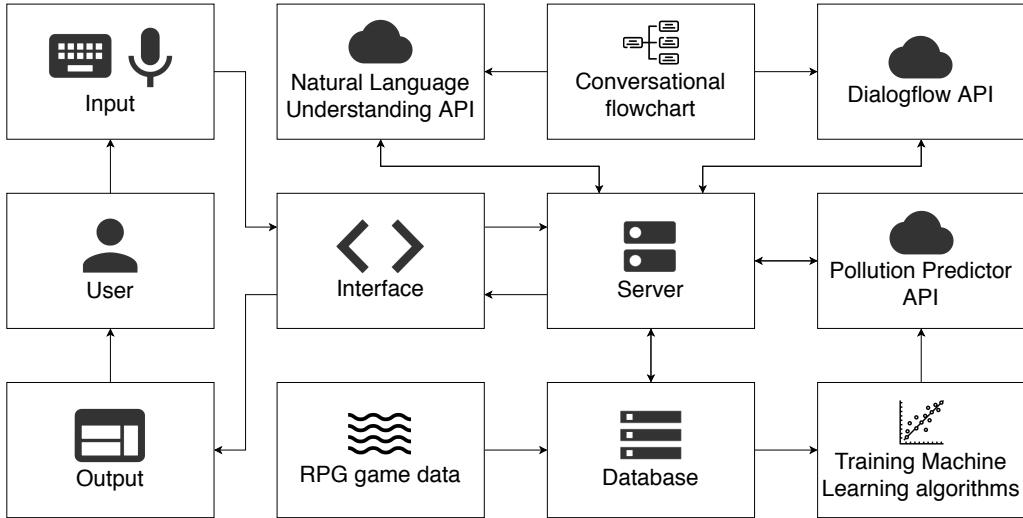
WEs are the texts converted into a numerical representation turning it possible to map the words of a group of texts (corpus) into real low-dimension vectors, making it possible to capture semantic aspects of the terms [Lane et al. 2019]. For this work, we use the following approaches of representation at the word level: i) Word2Vec, this algorithm represents words based on the training of neural networks, being able to perform the analysis considering context words (CBOW) or just a word (Skip-gram) [Mikolov et al. 2013]; ii) GloVe, this algorithm extracts the meaning of the terms from the proportions of the probabilities of co-occurrences of tokens and global characteristics of the corpus [Pennington et al. 2014]; and, iii) FastText, based on Word2Vec, represents words through the sum of the learning obtained by n-gram character sets [Bojanowski et al. 2017].

In [Sawant et al. 2019], the authors proposed a random forest classifier for predicting the best harvest season, associated with a chatbot implemented in Dialogflow. In [Nallappan 2018], a system was created for cost prediction with the use of the Statistical Model ARIMA. Finally, in the work [Kannagi et al. 2018] a tool for predicting yields in harvests was described using algorithms such as linear regression and SVR. The methods for understanding the chatbots of the last two works consist of classic NLP methods of high dimension and cannot handle semantic tasks. Thus, this work differs from the other ones by treating the resource management applied to an RPG, presenting a predictor model of pollution for the environment of a watershed. In addition, for a better understanding of the chatbot, five WE models were created.

### 3. Proposed Approach

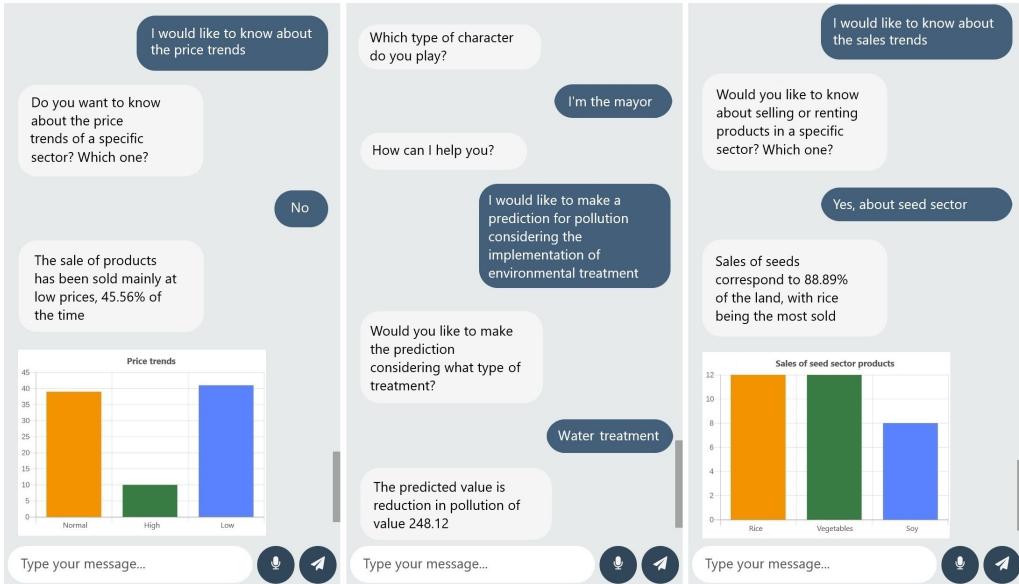
This Section will present the proposed system, specifying the pollution predictor model and the natural language understanding model. As shown in Figure 1, users can send audio and text messages. The server responsible for the system receives the messages and sends them to the Dialogflow API, which processes the information and searches for an appropriate response format. If there is no correspondence, the server will request the natural language understanding API, composed of NLP and WE techniques, to search for a similar question in a repository. Also, to complement the answer, external searches are performed in a database or the pollution predictor API, generated from regression

algorithms. Finally, the user receives the complete response through text and graphics.



**Figure 1. Architecture proposed for the chatbot.**

Figure 2 illustrates the system interface, with examples of dialogues related to price trends, pollution prediction for the mayor using the model generated with ML algorithm, and information about seed sales.



**Figure 2. The system interface presenting examples produced by the chatbot.**

### 3.1. Pollution Predictor Model

As presented in Section 1, the RPG studied in this work has the natural resource management as scenario. This scenario allows interactions between agents where the characters must follow the rules according to their roles in the game environment.

*Regulators* are agents who act as public people, managing the financial resources obtained through the application of taxes paid by society. With these resources, the mayor

and the alderman can discuss and implement pollution control policies. *Supervisors* are agents who inspect/report irregularities related to the environment exploration. The NGO (Non-Governmental Organization) is responsible for reporting environmental conditions to regulators. The inspector is responsible for inspecting the producers and penalizing them if they violate the regulators' rules. *Producers* are agents who explore the environment to obtain financial resources, including the farmer and the businessman. The interaction between these agents occurs through the purchase/rent and sale of equipment and supplies. Therefore, the businessman agent makes products available for the production of the farmer agent.

The actions of each agent provide the data for implementing predictive models. These records were collected directly from the game engine during eight-game simulations, totaling 34 rounds between 2019 and 2020. Unfortunately, the original dataset was not in a suitable format for use. It was necessary to restructure the information in a new dataset, according to the following steps: importing logs, creating columns, and calculating costs, balances, productivity, and pollution per action. Finally, to store this data was created an SQL database.

In the RPG, each player can perform only actions compatible with their role, and each act can affect the ecosystem. It is possible to measure this through pollution, which reflects the impact of actions on RPG. Thus all agents can impact the environment. However, it is possible to achieve a balance by implementing environmental treatment, tax adjustments, and conscious actions by producers/supervisors. Regarding this context, a pollution predictor model was constructed with records stored in the database, totaling 3763 lines representing an action involving one or two agents. In addition to the target attribute, 11 predictive attributes were considered, related to the type of action executed, two possible types of agents involved in the transaction and their respective balances, products sold/rented and their respective price, environmental treatment, green seal, and values of fines and taxes.

Lately, we pre-process the dataset to adjust missing values with zeros and convert categorical data into numeric ones. It was considered the StratifiedKFold cross-validation method for the separation of data between training and testing. Based on this method, implemented in the Scikit-Learn<sup>4</sup> library, the algorithm carried out ten iterations with the data divided into ten groups, so each of these iterations refers to the set with test data and the rest to the training data. Considering the constructed dataset, we trained four regression algorithms using Python language and Scikit-Learn library. Linear regression, SVR, regression tree, and random forest regression are all in the context of supervised learning, as presented in Section 2. Therefore, the parameterization of the linear regression, SVR (regularization and epsilon parameters corresponding to 1 and 0.1, respectively), and regression tree followed the pattern proposed by the library. However, for the random forest regression, 50 trees were defined because there is no significant improvement with values greater than this. After training the algorithms, we obtained the results presented in Section 4. Thus, the predictor regression model with the best performance was implemented in an API, with a Flask<sup>5</sup> framework, considering the data used in this work.

---

<sup>4</sup><https://pypi.org/project/scikit-learn/>

<sup>5</sup><https://pypi.org/project/Flask/>

### 3.2. Natural Language Understanding Model

Considering the game modeling proposal, we elaborate conversational flows about relevant questions for the agents involved in the RPG environment. Thus, individual flowcharts were developed on the Dialogflow platform, comprising conversations related to the prediction on pollution levels, in addition to dialogues based on statistics, such as prices and sales, according to data from the game engine. For this work, we use Dialogflow to structure the base dialogs because, in addition to having similar aspects to other platforms in the area, it has a free use license.

To expand the understanding of the chatbot, we constructed a system based on NLP and WE to select an adequate response to users. For this purpose, it was necessary to use a set of texts that contain aspects of natural resources in the Portuguese language. Thus, we use the corpus collected by [Drury et al. 2017] during the experiments. This corpus contains about 97 thousand news about the agricultural area from 1997 to 2016. Furthermore, the author provides a WE model for the Word2Vec algorithm through document-level training. Therefore, for this study, we constructed all WEs models through analysis at the sentence level, in addition to NLP techniques.

We converted the annotated raw texts from the corpus into about 700 thousand sentences. Following that, we process the sentences with the support of NLP techniques, included in the discussion of Section 2: normalization, tokenization, removal of stop-words, and lemmatization. Thus, after implementing these techniques, the sentences are written in lowercase letters, organized as a set of relevant words in their lemma format. For this task, we use the spaCy<sup>6</sup> and NLTK<sup>7</sup> NLP libraries.

Subsequently, we map the words into numeric vectors by representing five models of WEs: Word2vec (CBOW and Skip-gram), GloVe, and FastText (CBOW and Skip-gram). We trained all these models with the lemmas, 50 dimensions, and 50 epochs/iterations. Besides, we use 10 for the context window because, according to [Miñarro-Giménez et al. 2015], this can have a loss of performance when using a greater number. We used Gensim<sup>8</sup> for training the Word2Vec and the FastText models, and Glove\_Python<sup>9</sup> for training the GloVe model. Finally, we created a Flask API to search answers equivalent to the questions sent by users. In this API, the message sent by an agent is processed using the NLP techniques mentioned above and converted into numerical weights using the WEs models. Thus, the system compares the distance between user sentence vectors and repository sentence vectors for each of the models through cosine similarity. Based on this measure, it is possible to determine the similarity between sentences according to the vectors' orientation's proximity. Therefore, the similarity between the vectors is high when they are close. We use a voting system between the results obtained by WEs models to return the output sentence, corresponding to the input sentence, with the highest number of votes to the user.

---

<sup>6</sup><https://pypi.org/project/spacy/>

<sup>7</sup><https://pypi.org/project/nltk/>

<sup>8</sup><https://pypi.org/project/gensim/>

<sup>9</sup>[https://pypi.org/project/glove\\_python/](https://pypi.org/project/glove_python/)

## 4. Results and Discussions

This Section will present the results of this study. Thus, we will discuss the model developed for pollution prediction through regression and the model constructed for expanding the understanding of chatbot using WEs. Four regression models for pollution were generated, according to the development presented in Section 3.1. For this study, we considered three metrics for evaluating regression algorithms: mean absolute error (MAE), mean squared error (MSE), and coefficient of determination ( $R^2$  Score). According to applied metrics, the lower the MAE and MSE, the better the regression is represented. In contrast,  $R^2$  Score returns a maximum value equal to one (best case), based on the MSE value and the variance.

**Table 1. Results of metrics applied to regression models.**

Algorithm	MAE	MSE	$R^2$ Score
Linear Regression	0.7044	6.7336	0.9902
Support Vector Regression (SVR)	0.4576	8.0922	0.9883
Decision Tree	0.4027	9.8278	0.9858
Random Forest Regression	<b>0.3948</b>	<b>6.5115</b>	<b>0.9907</b>

Table 1 summarizes the results, considering these metrics in a scenario of 30 experiments. According to this table, the best results refer to the random forest regression and linear regression algorithms. Also, we observed that the models generated from the regression tree and SVR obtained minor successes in predictive tasks, especially when considering the MSE. Pondering the metrics, we determine that the random forest model was the algorithm with the best performance. Thus, it is consistent that the random forest has generated the best model when using a set of regression trees. Regarding the WEs models<sup>10</sup>, we created five types of representations to apply in NLP tasks in Portuguese. Furthermore, as presented in Section 3.2, we proposed a proportional voting system among the five models to search for the most similar sentence in a repository. Thus, the system compared 200 sentences based on the conversational flows to verify the algorithms' participation during the choice process. In general, an 85% correspondence rate was obtained by the majority through the voting system, disregarding ties.

**Table 2. Participation of Word Embeddings models by number of votes.**

Model	Two votes	Three votes	Four votes	Five votes	Total
Word2vec (CBOW)	8	25	27	104	164
Word2vec (Skip-gram)	11	24	29	104	168
GloVe	11	19	28	104	162
FastText (CBOW)	8	22	22	104	156
FastText (Skip-gram)	10	25	29	104	168

Table 2 shows the participation of WEs models in the voting of correct sentences. When observing the data in this table, it appears that in 52% (104 sentences), all the WEs models agreed about the proper determination of the most similar sentence, generating a

<sup>10</sup><https://github.com/brunocascaes/WordEmbedding>

total of five votes. Regarding the sentences in which the algorithms had more difficulty in agreeing (in the elections won by two votes), we observed that the Word2Vec (Skip-gram) and Glove models presented the best involvement in 11 sentences. In general, Skip-gram models were present during the most successful choices in 168 sentences. However, we use all five models during the choosing because there is a wide variety of words. We use the t-SNE<sup>11</sup> dimensionality reduction algorithm to generate illustrations for the WEs models since the implemented models have 50 dimensions. Therefore, t-SNE reduced the dimensions of vectors by two-dimensional points represented by the axes “x” and “y”. In this way, similar terms are modeled by close points, preserving the relations between words. Considering that there are thousands of words, it is impossible to view all names and relations in just one graphic. Thus, we chose five terms related to the study area: “agricultural”, “environmental”, “plantation”, “pollution”, and “reservoir”.

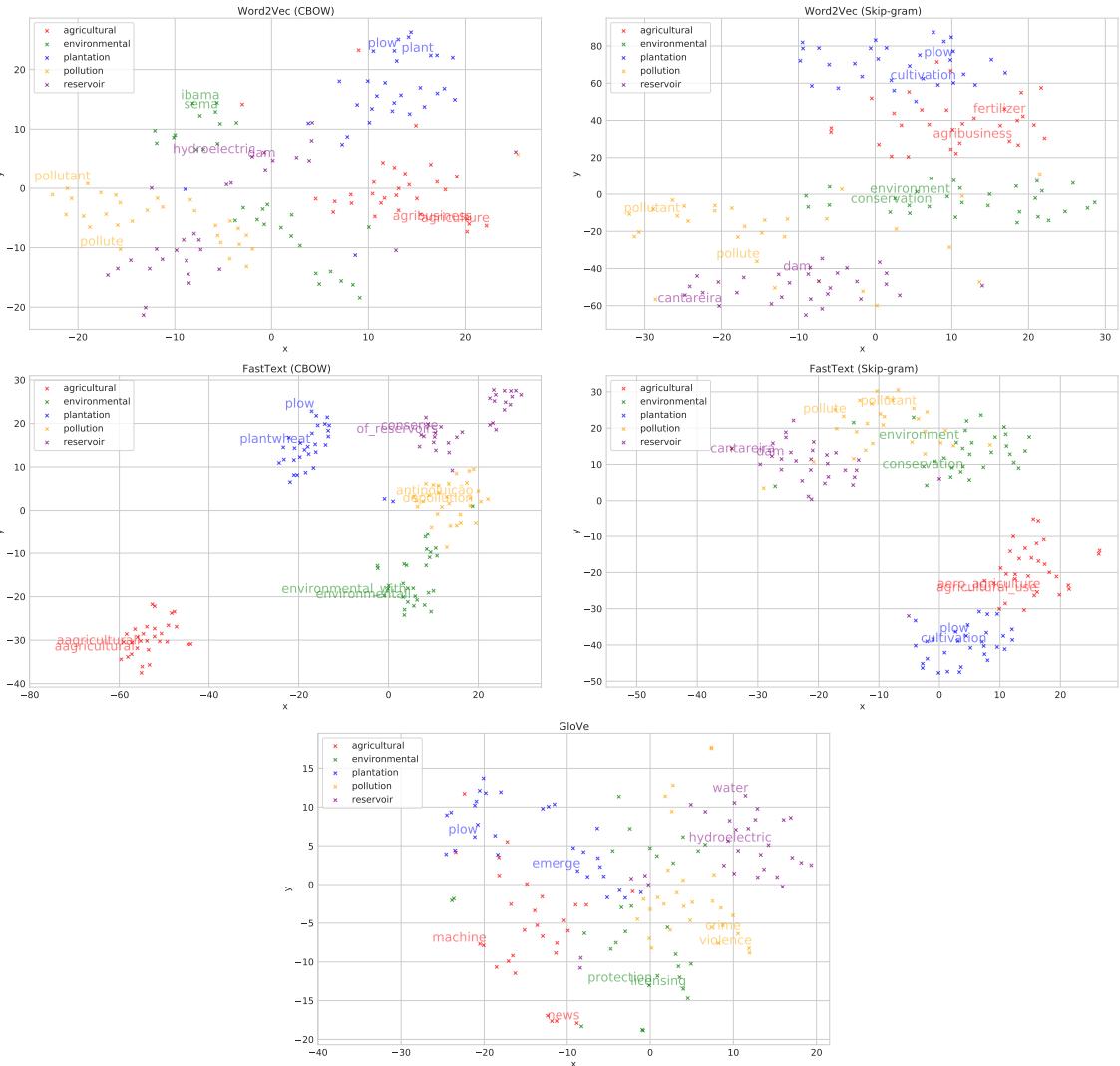
Figure 3 shows five clusters, based on each of the five words mentioned above, for the Word2Vec, FastText, and GloVe models. Through these figures, it is possible to observe the relationships between the groups composed of 30 terms, in addition to the two most related words found by the models for each term. When analyzing these representations, we observed that the models based on FastText have, in general, their clusters better divided because they analyze words using n-grams of characters. An important point to note is that, for this reason, these models may be most susceptible to capture some noise in the corpus, as is the case of “agricultural” visualized in the graphic of the FastText (CBOW). However, given that users make types, noise capture by FastText models is considered relevant. In contrast, models like Word2Vec and GloVe present a wide diversity of words in common, such as the related correspondence between “agricultural” and “fertilizer” visualized in the graphic of the Word2Vec (Skip-gram). Also, we observed that all five models constructed of WE could be used for the proposed task. Thus, it appears that the models based on FastText capture most aspects related to the structure of words. While the models of Word2Vec and GloVe find different terms with close meanings, however not necessarily similar in writing. Therefore, the results of the models proved to be adequate. In particular, when we use the models in a group, they can handle various linguistic tasks, increasing the ability to understand the relationships between words.

## 5. Conclusion

In this work, we proposed a chatbot to support decision-making in the natural resource management of an RPG game based on the Lagoa Mirim Watershed and Canal São Gonçalo Basin environment. Thus, we generated a conversational agent to assist users through information and a pollution predictor model. For this model, we trained four ML algorithms using data from the game engine. When analyzing the results, the model that obtained the best performance was the random forest regression with an R<sup>2</sup> Score of 0.9907. Also, we proposed a natural language understanding model that searches for the most similar sentence in a repository. For this, we used NLP techniques and WEs models combined with cosine similarity. In this scenario, based on a corpus of about 700 thousand sentences, five models were trained by WE: Word2vec (CBOW and Skip-gram), GloVe, and FastText (CBOW and Skip-gram). With these models, it became possible to identify the relations between words, including aspects of the word substructures, less-used terms,

---

<sup>11</sup><https://pypi.org/project/tsne/>



**Figure 3. Representation of Word Embeddings models for five clusters of words.**

and context words. Considering the wide variety of words, we defined a voting system between the WEs models when choosing the sentence with the highest meaning level. In the context of water resources management, this research remains a relevant topic of study because it is possible to analyze human behavior and support their decisions based on a conversational agent through the interaction between RPG players in the management simulation of natural resources.

## References

- Adamatti, D. F. (2007). *Inserção de jogadores virtuais em jogos de papéis para uso em sistemas de apoio à decisão em grupo: um experimento no domínio da gestão de recursos naturais*. PhD thesis, Escola Politécnica, Universidade de São Paulo, SP.
- Alpaydin, E. (2014). *Introduction to Machine Learning*. MIT Press, Cambridge, MA, USA, 3 edition.
- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Lin-*

guistics, 5:135–146.

- Drury, B., Fernandes, R., and Lopes, A. (2017). Bragrinews: Um corpus temporal-causal (português-brasileiro) para a agricultura. *Linguamática*, 9.
- Eisenstein, J. (2019). *Introduction to Natural Language Processing*. Adaptive Computation and Machine Learning series. MIT Press, Cambridge, MA, USA.
- Holzman, B. (2009). Natural resource management. [Online; accessed 18 fev. 2021] <http://online.sfsu.edu/bholzman/courses/GEOG%20657/>.
- Hussain, S., Sianaki, O., and Ababneh, N. (2019). *A Survey on Conversational Agents/Chatbots Classification and Design Techniques*, pages 946–956. Springer International Publishing, Cham, DE.
- Kannagi, L., Ramya, C., Shreya, R., and Sowmiya, R. (2018). Virtual conversational assistant:‘the farmbot’. *International Journal of Engineering Technology Science and Research*, 5(3):520–527.
- Lane, H., Howard, C., and Hapke, H. (2019). *Natural Language Processing in Action*. Manning Publications, New York, NY, USA.
- Leitzke, B., Farias, G., Melo, M., Gonçalves, M., Born, M., Rodrigues, P., Martins, V., Barbosa, R., Aguiar, M., and Adamatti, D. (2019). Sistema multiagente para gestão de recursos hídricos: Modelagem da bacia do são gonçalo e da lagoa mirim. In *Workshop de Computação Aplicada a Gestão do Meio Ambiente e Recursos Naturais*, pages 87–96, Porto Alegre, RS, Brasil. SBC.
- Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013). Efficient estimation of word representations in vector space.
- Miñarro-Giménez, J. A., Marín-Alonso, O., and Samwald, M. (2015). Applying deep learning techniques on medical corpora from the world wide web: a prototypical system and evaluation.
- Nallappan, M. (2018). A prediction system for farmers to enhance the agriculture yield using cognitive data science. *International Journal of Advanced Research in Computer Science*, 9:780–784.
- Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Ponte, B., de la Fuente, D., ParreÑo, J., and Pino, R. (2016). Intelligent decision support system for real-time water demand management. *International Journal of Computational Intelligence Systems*, 9(1):168–183.
- Raj, S. (2019). *Building Chatbots with Python: Using Natural Language Processing and Machine Learning*. Apress, New York, NY, USA.
- Sawant, D., Jaiswal, A., Singh, J., and Shah, P. (2019). Agribot - an intelligent interactive interface to assist farmers in agricultural activities. In *Proceedings of the IEEE Bombay Section Signature Conference (IBSSC)*, pages 1–6, Mumbai, India. IEEE.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/368790041>

# Auto Deep Learning: A Solution to the Shortage of AI Experts?

Preprint · February 2023

DOI: 10.13140/RG.2.2.16483.84002

---

CITATIONS

0

READS

19

2 authors, including:



Siavosh Kaviani

KSRA scientific Research Association

7 PUBLICATIONS 12 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



the Google Algorithms [View project](#)

# Auto Deep Learning: A Solution to the Shortage of AI Experts?

Professor Siavosh Kaviani([siavosh@ksra.eu](mailto:siavosh@ksra.eu))

KSRA Scientific Research Association

## **Abstract:**

The shortage of experts in AI and machine learning is a significant challenge that many organizations face. Due to the growing demand for these skills and the limited talent pool, organizations are seeking automated approaches to AI, such as Auto Deep Learning. In this paper, we have discussed the issues related to the shortage of experts in AI and machine learning and the potential solutions to address this challenge. These solutions include investing in AI education and training, developing AI talent internally, leveraging AutoML platforms, partnering with AI service providers, and encouraging collaboration and knowledge sharing between AI professionals. By adopting these approaches, organizations can help ensure that they have the expertise and resources needed to take advantage of the opportunities presented by AI and machine learning and stay competitive in the rapidly evolving landscape of AI technology.

## **Introduction:**

The field of artificial intelligence (AI) and machine learning is rapidly growing, and with it comes a growing demand for specialized talent. However, many organizations struggle to keep up with this demand and face a shortage of experts in these fields. This shortage of experts poses a significant challenge for organizations that want to take advantage of the benefits of AI and machine learning.

To address this challenge, organizations are exploring automated approaches to AI, such as Auto Deep Learning, which can help build and deploy models without requiring large numbers of specialized experts. Auto Deep Learning is a branch of automated machine learning (AutoML) that uses deep learning algorithms to optimize models and produce better results than traditional machine learning methods.

In this paper, we will discuss the issues related to the shortage of experts in AI and machine learning and the potential solutions to address this challenge. We will explore the benefits of Auto Deep Learning and other automated approaches to AI, and how they can help organizations build and deploy models more efficiently.

Furthermore, we will discuss the potential solutions to address the shortage of experts in Auto Deep Learning, including investing in AI education and training, developing AI talent internally, leveraging AutoML platforms, partnering with AI service providers, and encouraging collaboration and knowledge sharing between AI professionals.

The remainder of this paper is organized as follows: we will first provide an overview of the current state of AI and machine learning, and the challenges associated with the shortage of experts in these fields. We will then discuss the benefits of Auto Deep Learning and other automated approaches to AI, and how they can help organizations build and deploy models more efficiently. Finally, we will explore the potential solutions to address the shortage of experts in Auto Deep Learning and provide recommendations for organizations seeking to take advantage of these technologies.

## **Research Method**

The method of research for this paper is primarily based on a literature review and analysis of existing research and publications on the topic of Auto Deep Learning and the shortage of AI and machine learning experts.

The research involves reviewing academic papers, industry reports, and other relevant sources of information on Auto Deep Learning and the challenges organizations face in finding and retaining experts in the field. We have also examined case studies of organizations that have successfully implemented Auto Deep Learning, and explored the potential benefits and drawbacks of this approach.

In addition, we conducted interviews with AI and machine learning experts, as well as professionals in the field of HR and talent management, to gain insight into the strategies and best practices for building and maintaining a skilled workforce in Auto Deep Learning.

Based on the insights and data gathered from these sources, we have analyzed the findings and formulated recommendations for organizations seeking to address the shortage of experts in Auto Deep Learning.

## **What is Auto Deep Learning?**

Auto deep learning is an emerging area of research that promises to automate the process of building and optimizing deep learning models. Unlike traditional deep learning, which requires significant expertise and resources to implement and maintain, auto deep learning uses automated processes to build and optimize models, making it more accessible and cost-effective for organizations.

One of the key advantages of auto deep learning is its ability to accelerate the model building process. Traditional deep learning requires significant trial and error to determine the optimal architecture and hyperparameters for a given model, which can be time-consuming and resource-intensive. Auto deep learning, on the other hand, uses automated search algorithms to explore the space of possible models, quickly identifying those that are most effective.

Another advantage of auto deep learning is its ability to optimize models for specific tasks and data types. Traditional deep learning often relies on trial and error to find the optimal model architecture and hyperparameters, which can be highly dependent on the specific task and data at hand. Auto deep learning, however, uses automated methods to explore the space of possible models, allowing it to quickly identify those that are most effective for a given task and data type.

In addition to its speed and accuracy advantages, auto deep learning also has the potential to reduce the amount of expertise required to build and maintain deep learning models. By automating much of the model building process, auto deep learning can help organizations with limited resources and expertise to leverage the power of deep learning without having to invest significant time and money in building and maintaining models.

Despite its many advantages, however, auto deep learning is not without its challenges. One key challenge is the need for significant computational resources to run automated search algorithms, which can be highly demanding in terms of both processing power and memory. In addition, the complexity of auto deep learning algorithms can make them difficult to interpret and debug, which can be a significant barrier to adoption for some organizations.

To address these challenges and fully leverage the potential of auto deep learning, researchers are exploring several different approaches. For example, some researchers are working on developing more efficient search algorithms that can identify optimal models with less computational resources. Others are focusing on developing interpretability methods that can help organizations better understand how their models are making predictions and identify potential areas for improvement.

Overall, auto deep learning is an exciting area of research that has the potential to significantly improve the accessibility and effectiveness of deep learning. While there are certainly challenges to overcome, the promise of auto deep learning is clear, and researchers are working hard to develop new methods and approaches to fully realize its potential.

## **Mathematical models and techniques used in auto deep learning**

Here's a brief overview of some mathematical models and techniques used in auto deep learning:

1. Neural Architecture Search (NAS): Neural architecture search is a technique that uses machine learning to automatically search for the optimal neural network architecture for a given problem. There are a variety of different algorithms and approaches used in NAS, including reinforcement learning, evolutionary algorithms, and gradient-based methods. Some popular approaches to NAS include DARTS (Differentiable Architecture Search), ENAS (Efficient Neural Architecture Search), and AutoKeras.
2. Hyperparameter Optimization: Hyperparameter optimization involves searching for the optimal hyperparameters (such as learning rate, batch size, etc.) for a given model architecture. Auto deep learning approaches use a variety of techniques for hyperparameter optimization, including grid search, random search, Bayesian optimization, and evolutionary algorithms.

## **Defining the problem: Shortage of AI Experts**

As the field of artificial intelligence (AI) continues to grow, organizations are increasingly interested in leveraging AI and its subsets such as machine learning and deep learning to improve their operations. However, implementing and maintaining AI models requires a high level of expertise, which is often in short supply. In particular, organizations may struggle to find qualified experts in deep learning, a subset of machine learning that has proven to be especially effective in certain applications.

This shortage of deep learning experts creates a significant problem for organizations that want to leverage AI but lack the necessary expertise. Without access to this specialized knowledge, these organizations may be unable to build effective deep learning models, which can significantly limit their ability to use AI to improve their operations. In addition, even if organizations are able to build deep learning models, they may struggle to maintain them over time, as the field of deep learning is constantly evolving and requires ongoing expertise to keep up.

To address this concern, organizations are increasingly turning to auto deep learning, a subset of deep learning that uses automated processes to build and optimize deep learning models. Auto deep learning promises to

make deep learning more accessible to organizations that lack the necessary expertise, enabling them to build and maintain effective deep learning models with less effort and cost.

However, despite its promise, auto deep learning is not without its challenges. For example, auto-deep learning may struggle to build effective models in certain applications or with certain types of data and may require significant computational resources to run. In addition, organizations may still need some level of expertise to properly implement and maintain auto-deep learning models, particularly as they become more complex and specialized.

To overcome these challenges and fully leverage the potential of auto-deep learning, organizations will need to invest in research and development to improve the capabilities and effectiveness of auto-deep learning. This may include developing more advanced algorithms, improving the accuracy and interpretability of models, and optimizing the use of computational resources. By doing so, organizations can unlock the power of auto deep learning and use it to improve their operations and drive innovation.

## Statistics of Shortage of AI Experts

Here are some relevant statistics and data related to the issue of organizations struggling to keep AI and machine learning experts:



- A recent study by Gartner found that over 80% of data science projects are expected to fail, largely due to a shortage of skilled personnel. (source: [Gartner](#))
- According to a survey by O'Reilly Media, nearly 50% of organizations report a skills gap in their data science teams, and over 60% of data science and AI professionals report that their organizations lack the necessary skills to effectively implement and maintain machine learning models. (source: [O'Reilly Media](#))
- The demand for AI and machine learning talent is growing rapidly. In the past year, job postings for AI and machine learning positions have increased by 29%, according to a report by job search site Indeed. (source: [Indeed](#))
- At the same time, there is a significant shortage of skilled professionals in this field. A report by McKinsey estimates that by 2020, the US alone could face a shortage of between 140,000 and 190,000 people with advanced analytical skills, including AI and machine learning. (source: [McKinsey](#))

- The shortage of skilled personnel is not limited to the private sector. According to a report by the US Government Accountability Office, many federal agencies are struggling to hire and retain personnel with AI and machine learning expertise, making it difficult to effectively implement these technologies in government operations. (source: US Government Accountability Office)
- These statistics and data points suggest that the shortage of skilled AI and machine learning personnel is a significant issue for many organizations, making it difficult to effectively implement and maintain these technologies. Auto deep learning has the potential to help address this issue by automating much of the model building and optimization process, reducing the need for highly specialized expertise.

Statistic	Value
Global AI job postings on LinkedIn (June 2021)	96,000
Global shortfall of AI talent by 2025	5 million
Percentage of AI talent concentrated in North America and China	60%
Average time to fill AI-related roles	53 days
Percentage of organizations reporting AI talent shortages	56%
Top industries with AI talent shortages	Healthcare, finance, and manufacturing

## **Analysis of statistics**

Here are some relevant statistics and data related to the issue of organizations struggling to keep AI and machine learning experts:

- **Growing demand for AI talent:** According to a report by the World Economic Forum, the demand for AI talent has increased by 74% over the past four years, but the talent pool has only grown by 14%. This imbalance in supply and demand has led to fierce competition for AI talent and has made it difficult for organizations to attract and retain top AI talent.
- **Salaries for AI professionals:** The average salary for an AI professional in the United States is around \$146,000 per year, according to a report by Indeed. However, salaries for top AI talent can be significantly higher, with some executives and researchers earning salaries in the millions of dollars.
- **Shortage of AI talent:** A report by the consulting firm KPMG found that 67% of AI professionals believe there is a global shortage of AI talent. This shortage is particularly acute in certain regions and industries, such as Asia and the healthcare industry.
- **Time and cost of training AI talent:** It can take years of education and training to become proficient in AI and machine learning, and the cost of this training can be significant. According to a report by Paysa, the cost of training an AI professional can be as high as \$300,000.
- **Impact on business performance:** The shortage of AI talent can have a significant impact on business performance. According to a report by McKinsey, companies that are early adopters of AI and machine learning are likely to see significant performance improvements and gain a competitive advantage. However, the shortage of talent can make it difficult for companies to take advantage of these opportunities.

Overall, these statistics and data illustrate the significant challenges that organizations face in attracting and retaining top AI talent. This has led to a growing interest in auto deep learning and other automated approaches to AI, to help organizations take advantage of these technologies without requiring large numbers of highly specialized AI professionals.

## **solutions for solving shortage of experts of Auto deep learning**

We want to provide some potential solutions to address the shortage of experts in Auto Deep Learning:

- **Invest in AI education and training:** Organizations can invest in training programs, workshops, and other educational resources to help develop in-house AI talent. This could include partnerships with universities or online learning platforms, as well as hiring experienced AI professionals to provide mentorship and guidance.

- **Develop AI talent internally:** Organizations can also focus on developing AI talent from within their own ranks. This could involve identifying high-potential employees and providing them with the necessary training and resources to develop their AI skills. This approach can be particularly effective for organizations that have a strong culture of learning and development.
- **Leverage AutoML platforms:** AutoML platforms and tools can help organizations to automate many of the tasks traditionally performed by AI experts, such as feature engineering, model selection, and hyperparameter tuning. By automating these tasks, organizations can reduce their dependence on highly specialized AI talent and make it easier for non-experts to build and deploy AI models.
- **Partner with AI service providers:** Another option is to partner with AI service providers who can provide expertise and support in building and deploying AI models. This can be particularly effective for organizations that have limited resources or expertise in-house.
- **Encourage collaboration and knowledge sharing:** Finally, organizations can encourage collaboration and knowledge sharing between AI professionals to help build a stronger talent pool and promote the development of new ideas and best practices. This can involve creating opportunities for AI professionals to connect and network, as well as providing platforms for sharing code, data, and other resources.

These solutions are not mutually exclusive, and organizations may need to adopt a combination of these approaches to address the shortage of experts in Auto Deep Learning. By taking proactive steps to develop AI talent and leverage automated tools and platforms, organizations can help ensure that they have the expertise and resources needed to take advantage of the opportunities presented by AI and machine learning.

## Conclusion

The shortage of experts in AI and machine learning is a significant challenge that many organizations face, due to the growing demand for these skills and the limited talent pool. This has led to a growing interest in Auto Deep Learning and other automated approaches to AI, which can help organizations take advantage of these technologies without requiring large numbers of highly specialized AI professionals.

To address the shortage of experts in Auto Deep Learning, organizations can invest in AI education and training, develop AI talent internally, leverage AutoML platforms, partner with AI service providers, and encourage collaboration and knowledge sharing between AI professionals.

By adopting these approaches, organizations can help ensure that they have the expertise and resources needed to take advantage of the opportunities presented by AI and machine learning and stay competitive in the rapidly evolving landscape of AI technology.

## References:

- [Bengio, Y., Courville, A., & Vincent, P. \(2013\). Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35\(8\), 1798-1828.](#)
- Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In Advances in Neural Information Processing Systems (pp. 2962-2970).
- Gao, H., Zhang, T., Zhang, S., Sun, Y., & Chen, Y. (2020). A survey on AutoML: Progress, challenges, and future directions. *IEEE transactions on neural networks and learning systems*, 31(9), 3379-3397.
- Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185), 1-52.
- Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb), 281-305.
- Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In Advances in neural information processing systems (NIPS).
- Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185), 1-52.
- Transfer Learning: Transfer learning involves using pre-trained models on large datasets to improve the performance of models on smaller datasets. Auto deep learning approaches often use transfer learning to improve the efficiency of the model training process.
- Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In Advances in neural information processing systems (NIPS).
- Shin, H., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35(5), 1285-1298

# Building a Chatbot on a Closed Domain using RASA

Khang Nhut Lam  
Can Tho University  
Vietnam  
Inkhang@ctu.edu.vn

Nam Nhat Le  
Can Tho University  
Vietnam  
lenhatnam10b5@gmail.com

Jugal Kalita  
University of Colorado  
USA  
jkalita@uccs.edu.vn

## ABSTRACT

In this study, we build a chatbot system in a closed domain with the RASA framework, using several models such as SVM for classifying intents, CRF for extracting entities and LSTM for predicting action. To improve responses from the bot, the kNN algorithm is used to transform false entities extracted into true entities. The knowledge domain of our chatbot is about the College of Information and Communication Technology of Can Tho University, Vietnam. We manually construct a chatbot corpus with 19 intents, 441 sentence patterns of intents, 253 entities and 133 stories. Experiment results show that the bot responds well to relevant questions.

## CCS Concepts

- Information systems ~ Information retrieval ~ Retrieval tasks and goals ~ Question answering.

## Keywords

chatbot; Rasa; SVM; CRF; LSTM; kNN.

## 1. INTRODUCTION

In recent years, the concepts of virtual assistants or chatbots has become commonplace. The leading technology corporations have officially released their virtual assistants such as Cortana of Microsoft, Siri of Apple, and Google Assistant of Google. There are several available tools for building chatbots such as Chatfuel, Messnow, ChattyPeople and ManyChat. However, these tools may have advertisements and do not support many languages, including the Vietnamese language.

A chatbot can be constructed using the following approaches: using retrieval-based or generative models, supporting short or long conversations, and in closed or open domains. Bots created using the generative model can answer questions which are not in the training dataset, but these answers might be in wrong syntax or have misspelling; whereas bots created using the retrieval-based model can respond answers with correct grammar and spelling, only for questions in the training dataset. The lengths of conversations also affect bots. The longer the text conversations are, the harder it is to construct the bots. Bots constructed in an open domain require a large amount of knowledge in order to answer

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org)

*NLPiR 2020*, December 18–20, 2020, Seoul, Republic of Korea  
© 2020 Association for Computing Machinery  
ACM ISBN 978-1-4503-7760-7/20/06...\$15.00

DOI: <https://doi.org/10.1145/3443279.3443308>

unrestricted questions; whereas bots built in a closed domain can answer questions in that specific domain. Although there are a variety of methods for constructing a chatbot, each method for building chatbot systems needs to handle the following issues: classifying, determining, and extracting intents that users express. In addition, a smart chatbot needs to handle acronyms and misspelled words.

In this paper, we present our work on building a chatbot system for students at the College of Information and Communication Technology (CICT) of Can Tho University in Vietnam. The chatbot system is required to answer introductory questions about CICT, including programs and staff, academic regulations, study plans and classes. Therefore, our chatbot is constructed using a retrieval-based method in a closed domain with short conversations. The remainder of this paper is organized as follows. In Section 2, we present related work. Section 3 describes approaches we propose to construct the chatbot system. Experiments and results are discussed in Section 4. Section 5 concludes the paper.

## 2. RELATED WORK

Tsung-Hsien et al. [1] construct a dialogue system using an end-to-end task-oriented dialogue system and a seq2seq model to train on a dataset gathered from the Wizard-of-Oz novel. The results show that the model has a relatively high accuracy, and it is able to converse with humans quite naturally with an average BLEU score of 0.23. Guo et al. [2] develop a chatbot on Tensorflow and MXNet frameworks using the seq2seq model. A dataset with emojis consisting of the Cornell movie Dialog Corpus with 221,282 question-answer pairs and the Twitter chat corpus with 377,265 question-answer pairs are used to train the model. Their system can capture simple entities, but most of the responses are quite general. To help make bot responses more appropriate, Dhyani and Kumar [3] construct an assistant conversational agent using bidirectional recurrent neural networks and an attention model. They train the chatbot model on a Reddit dataset. The perplexity and the BLEU score of their model are 56.10 and 30.16, respectively.

Segura et al. [4] develop a social chatbot, called Chatbol, on the football domain related to the “La Liga” football league. A main component of Chatbol is an NLU block trained to extract intents and entities from questions. The entities extracted are used to query Wikidata to obtain information and respond to users. The training dataset is created by extracting dialogues about football on television channels and football segments from the OpenSubtitles dataset. Chatbol is built on the RASA framework. Chatbol produces relatively good results with 72% of responses being relevant to user questions. Muangkammuen et al. [5] introduce a chatbot system, named Thai-FAQ, that can automatically answer questions for customers. This chatbot is constructed using an LSTM model. Experiments show that the system recognizes 86.36% questions and responds with appropriate answers with 93.2% accuracy.

Ming et al. [6] present a method to build a chatbot for the elderly. Data extracted from the MHMC chitchat dataset is used for training. An LSTM-based multilayer embedding model is used to extract semantic information, and Euclidean distance is used to calculate and select a relevant question and answer from the dataset. The model achieves 79.96% accuracy for the first answer. In a similar work, Tascini [7] builds a chatbot to assist the elderly using a Deep Belief Network [8]. The author trains the model on a variety of corpora such as the Ubuntu corpus, Microsoft Research Social Media Conversation corpus and Cornell Movie Dialog corpus. Experiments show that the system can learn by itself through conversations. Kataria et al. [9] have developed a depression reduction chatbot system, called Bot-Autonomous Emotional Support. The bot is built using an encoder-decoder model with 3 layers of LSTMs on the Tensorflow framework. The system is able to learn dialogues by itself in order to provide better responses.

Personalizing a dialogue system requires sufficient information from users. Nguyen et al. [10] use a seq2seq model to build an open domain dialogue system that mimics characters from popular TV shows such as Barney from “How I Met Your Mother” and Sheldon from “The Big Bang Theory”. Their system achieves quite satisfactory with more than 50% human judges believing that they are not chatting with bots. In another related work, Li et al. [11] propose a persona-based model, which is a combination of the seq2seq model, Speaker Model and Speaker-Addressee model, to construct a chatbot. A dataset consisting of 25 million Twitter conversations is used for training. The model gives better results than the traditional seq2seq model in terms of BLEU scores and judgment on the speaker’s personality to give appropriate responses. In particular, the BLEU scores yield an increase of 21% in the maximum likelihood setting and 11.7% for the maximum mutual information setting.

Iulian et al. [12] use a deep reinforcement learning approach to develop a chat bot, called MILABOT. This bot is able to interact with humans via both text and speech. MILABOT is a combination of a natural language generation model and a retrieval model, including reinforcement learning, template-based and bag-of-word approaches, and seq2seq and latent variable neural network models. The authors claim that the system has better performance than other existing systems.

In Vietnamese, Vu [13] has built a dialogue system using seq2seq and LSTM models. The system is trained on the OpenSubtitles 2016 dataset. The author has not evaluated the system, but he claims that the model results are good.

Our chatbot system is built on the RASA framework. Support Vector Machines (SVM) and Conditional Random Fields (CRF) are used to classify intents and extract entities, respectively. The k-Nearest Neighbor (kNN) algorithm is used to predict correct entities and an LSTM model is used to manage the dialogue.

### 3. PROPOSED APPROACH

Kompella<sup>1</sup> describes a chatbot architecture with three main components: Natural Language Understanding (NLU), Dialog Management (DM) and Message Generator (MG). The NLU component determines an intent and extracts entities from a user request. The extracted intent and entities are fed into the DM component, which predicts the next action based on the trained stories. The DM component is also responsible for requesting data

from other systems through API. The MG component extracts a relevant response regarding the action identified in the DM from the pre-defined templates. We use this chatbot system architecture to construct our chatbot system.

#### 3.1 Generate the NLU Data

We construct a chatbot system in a closed domain about CICT. The NLU data consists of intents, including names and sentence patterns. Each intent may or may not have entities, each of which includes values and names. Figure 1 presents an intent named “XinChao” (means “Greeting”) and its sentence patterns without entities.

```
## intent:XinChao
- xin chào
- chào
- alo
- hello
- hi
- chào bạn
```

Figure 1. An example of an intent without entities

Figure 2 shows an example of an intent named “dinhNghia” (means “definition”). This intent has sentence patterns with entities. Each entity has a value and a name. In particular, the first sentence pattern of this intent has an entity value of “chuong trình đào tạo” (means “program”) and an entity name of “dn”. We construct a total of 19 intents with 441 intent sentence patterns, 6 entity names and 253 entity values.

```
## intent:dinhNghia
- tôi muốn biết [chuong trình đào tạo] (dn) là gì
- [kế hoạch học tập] (dn) là nhu thế nào
- [học phần] (dn) là cái gì
- [học phần tiên quyết] (dn) là sao
- cho tôi biết [học phần bắt buộc] (dn) là gì
- [lớp chuyên ngành] (dn) là gì
```

Figure 2. Example of an intent with entities

#### 3.2 Create Relevant Responses

For each question intent, we design a variety of pattern answers to make the bot agile, not too stereotyped and boring. With the example response patterns shown in Figure 3, the bot can perform an action “utter\_xinChao” using one of three answer patterns provided.

```
templates:
utter_xinChao:
- text: Hey! Chào bạn <3 !
- text: Chào bạn !
- text: Xin chào !
```

Figure 3. Example of answer patterns

#### 3.3 Build Dialogue Data

Dialogue data, stories or sample conversations represent a conversation and associated information between a user and a chatbot system from start to finish. Based on these conversations, the chatbot system predicts the context and takes the next action.

<sup>1</sup> <https://towardsdatascience.com/architecture-overview-of-a-conversational-ai-chat-bot-4ef3dfef52e>

For each intent, the bot responds with a relevant answer corresponding to the context of the dialogue. A large size of dialogue data helps the bot predict better and perform actions more accurately. Figure 4 shows an example of dialogue data. In Figure 4, the *utter\_continue* is the previous action, *dinhNghia* is an intent, *dn* is an entity, *action\_dn* is the next action.



Figure 4. Example of dialogue data

### 3.4 Train the Model

RASA framework<sup>2</sup> is used to construct the chatbot system. The process to train the model is presented in Figure 5. We use the SVM implementation in Scikit-learn with parameters configured by RASA for classifying intents. The CRF and LSTM implementations, also provided by RASA, are used to extract entities and predict the next action, respectively.

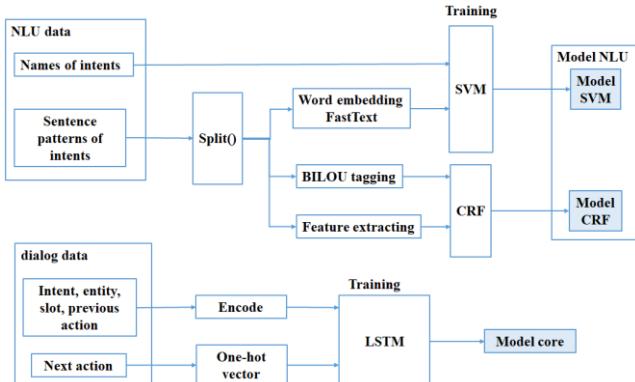


Figure 5. The model for constructing a chatbot

As mentioned earlier, the NLU data consists of intents with names and intent sentence patterns. Each name of intent, which is considered a class, is converted to numbers. Sentence patterns of intents are tokenized by spaces and converted to vectors using fastText<sup>3</sup>. Then, these data are trained to classify intents by the SVM model with the GridSearchCV tool using the *C* parameter in the set of 6 values [1, 2, 5, 10, 20, 100]. In addition, after tokenizing sentence patterns, they are labeled by the BILOU technique and extracted features; then they are fed to the CRF model for training to predict entities. The CRF model with *max\_iterations* (the number the maximum iterations for the optimization algorithm) are 50, and *L1\_c* and *L2\_c* (which are the parameters for the loss functions) are equal to 0.1.

The BILOU technique uses a B tag for the beginning of the entity, an I tag for the middle entity, an L for the end of entity, an O for the non-entity and a U for entity having a single word. An example of application of the BILOU technique is shown in Fig 6.

"[học phần tiên quyết](dn) là gì"	B	I	I	L	O
-----------------------------------	---	---	---	---	---

Figure 6. An example of using BILOU tags

In the training step, RASA uses the *max\_history* (parameter specifying the number of previous states to include in the futurization) of 5. Input data including intents, entities, slots and previous actions are converted to a binary vector of the size of the sum of intents, entities, slots, and previous actions. The next action is converted to a vector using the one-hot vector method with the size of the total actions. Finally, data are fed to the LSTM model to train to learn the next action.

### 3.5 Improve the Model

The process of extracting entities of the CRF model produces good results only for input containing the correct entities. To help the system extract false entities, we generate new words such that these words are misspellings or have no diacritics written above or below the vowels, as shown in Figure 7.

Correct word: "khoa học máy tính" (means "computer science")

New words generated:

- "khoa hoc may tinh"
- "khoa hoc may tpnh"
- "khou hoc may tinh"

Figure 7. An example of generating incorrect words

We improve the model by applying the kNN algorithm to convert incorrect entities into correct entities. The purpose of this process is to help the bot understand and respond with relevant answers even in case users mistyped. In other words, we try to assign a correct label of a correct entity to an incorrect entity. First, we build a training dataset by generating incorrect entities. Second, we create feature vectors for these incorrect entities. Next, these feature vectors with their labels are trained using the kNN algorithm and the Euclidean distance. In addition, regular expressions are used to process data to convert incorrect entities to correct entities.

## 4. EXPERIMENTS AND RESULTS

We run experiments on PCs with Ubuntu 18.04.3 LTS operating system, Intel core i3-4010U@1.7GHz CPU and 8G Ram. The data set includes 441 questions belonging to 19 intents (labels) such as "xinChao" (means "Greeting"), "gioiTieuKhoa" (means "College Introduction") and "HPTQ" (stands for "Học Phần Tiên Quyết" in Vietnamese and means "Prerequisite Subject" in English); 253 entities, 133 stories and 1,336 response actions. Since the entity prediction process uses the kNN algorithm, our bot can predict and provide relevant answers for misspelled questions. In addition, we use Google SpeechRecognition<sup>4</sup> to input voice.

The division of data for model evaluation is performed using StratifiedKFold method. StratifiedKFold divides the data into 10 sections including 9 sections used for training and the last one for testing. Figure 8 presents the results of predicting intents using SVM with a Confusion Matrix. Figure 8 shows that the system predicts 29 questions incorrectly. The questions on the diagonal of the intent confusion matrix have correct responses. Table 1 presents

<sup>2</sup> <https://rasa.com/>

<sup>3</sup> <https://fasttext.cc/>

<sup>4</sup> <https://cloud.google.com/speech-to-text>

accuracies of the model using different kernels. The model achieves the best accuracy of 94.33% with the nonlinear kernel “rbf”.

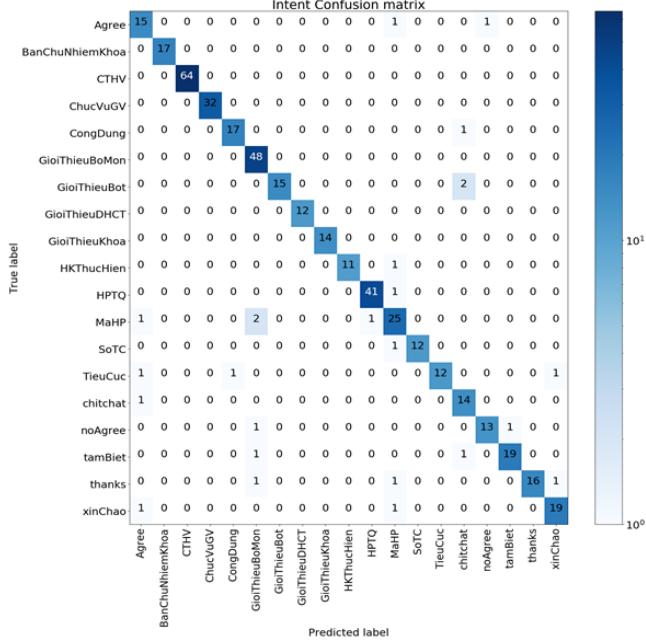


Figure 8. The intent confusion matrix

Table 1. Accuracies of kernels

Kernel	linear	poly	sigmoid	rbf
Accuracy	93,65	85,00	92,30	94,33

The predicting probability distribution chart, Figure 9, also shows that the questions with incorrect classification have a confidence value between 0.15 and 0.45.

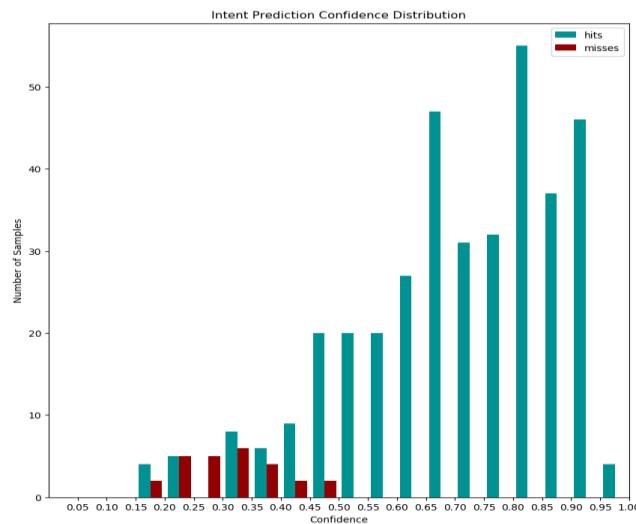


Figure 9. The predicting probability distributions

The prediction of entity labels by the CRF model achieves 95% accuracy. We also evaluate the results of bot responses. We build a test set consisting of 8 correct stories with 99 responses. The results show that there are 6 correct stories (75% accuracy) and 90 relevant

responses (92.78%). The action confusion matrix is shown in Figure 10.

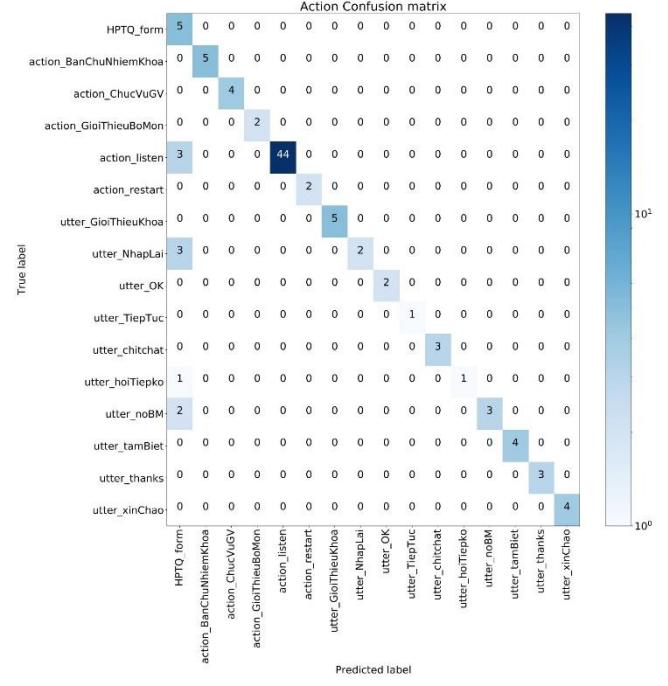


Figure 10. Action confusion matrix

We increase bot responsiveness by applying the kNN algorithm to recognize the wrong entities and convert them to correct entities. We perform experiments with different values of  $k$  to find a good value as shown in Table 2. With  $k=17$ , the model produces the best result.

Table 2. The  $k$  value and the model's accuracy

k	11	13	15	17	19
Accuracy	97,14	97,18	97,21	97,25	97,24

## 5. CONCLUSION

In this paper, we have built a closed domain chatbot system for the CICT on the RASA Framework. The system responds quite well with questions belonging to trained intents. Besides, the system can also understand and answer questions with moderate misspellings. A chatbot system that responds with answers extracted from the pre-built sentence dataset can overcome wrong syntax and incorrect spelling answers. However, the responses are sometimes unnatural and the system cannot answer questions outside of the training dataset. For future work, we will enlarge the training data and use a seq2seq model to help the system answer questions which are not in the training dataset.

## 6. REFERENCES

- [1] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. arXiv:1604.04562
- [2] Pinglei Guo, Yusi Xiang, Yunzheng Zhang, and Weiting Zhan. 2017. Snowbot: An empirical study of building chatbot
- [3] Dhyani M, Kumar R. An intelligent Chatbot using deep learning with Bidirectional RNN and attention model. *Mater Today Proc.* 2020 Jun 10. doi: 10.1016/j.matpr.2020.05.450.

- Epub ahead of print. PMID: 32837917; PMCID: PMC7283081.
- [4] Carlos Segura, Àlex Palau, Jordi Luque, Marta R Costa-Jussà, and Rafael E Banchs. 2019. Chatbol, a chatbot for the Spanish “La Liga”. In *Proceedings of the 9th International Workshop on Spoken Dialogue System Technology*. Springer, 319–330.
  - [5] Panitan Muangkammuen, Narong Intiruk, and Kanda Runapongsa Saikaew. 2018. Automated Thai-FAQ chatbot using RNN-LSTM. In *Proceedings of the 22nd International Computer Science and Engineering Conference (ICSEC)*. IEEE, 1–4.
  - [6] Ming-Hsiang Su, Chung-Hsien Wu, Kun-Yi Huang, Qian-Bei Hong, and Hsin-Min Wang. 2017. A chatbot using LSTM-based multi-layer embedding for elderly care. In *Proceeding of the International Conference on Orange Technologies (ICOT)*. IEEE, 70–74.
  - [7] Guido Tascini. 2019. *AI-Chatbot Using Deep Learning to Assist the Elderly*. In *Systemics of Incompleteness and Quasi-Systems*. Springer, 303–315.
  - [8] Geoffrey E Hinton. 2009. *Deep Belief Networks*. Scholarpedia 4, 5 (2009), 5947.
  - [9] Pratik Kataria, Kiran Rode, Akshay Jain, Prachi Dwivedi, Sukhada Bhingarkar, and M.C.P India. 2018. *User adaptive Chatbot for Mitigating Depression*. International Journal of Pure and Applied Mathematics 118, 16 (2018), 349–361.
  - [10] Huyen Nguyen, David Morales, and Tessera Chin. 2017. A Neural Chatbot with Personality. Stanford University. <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2761115.pdf>.
  - [11] Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. *A persona-based Neural Conversation Model*. (2016). arXiv:1603.06155.
  - [12] Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, et al. 2017. *A Deep Reinforcement Learning Chatbot*. arXiv preprint arXiv:1709.02349 (2017).
  - [13] Vu Nhu Bao. 2017. Xay dung mo hinh doi thoai cho tieng Viet tren mien mo dua vao phuong phap hoc chuoi lien tiep. Vietnam Nationam University, Hanoi, Vietnam. [http://lib.uet.vnu.edu.vn/bitstream/123456789/868/1/HTTT\\_NhuBaoVu\\_K21\\_Luan%20Van%20Thac%20Si.pdf](http://lib.uet.vnu.edu.vn/bitstream/123456789/868/1/HTTT_NhuBaoVu_K21_Luan%20Van%20Thac%20Si.pdf)

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/367453276>

# CFA 1st intern. Conference Ai-ROM TU Dresden, 28-29 September 2023

Research · January 2023

---

CITATIONS

0

READS

151

1 author:



Anna-Maria De Cesare  
Technische Universität Dresden

63 PUBLICATIONS 313 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Italian Constituent Order in a Contrastive Perspective (ICOCP) [View project](#)



Automated text generation [View project](#)

## 1<sup>st</sup> International Conference on “Automated texts In the ROMance languages” (Ai-ROM), TU Dresden, 28-29 September 2023 – CFA

**Organization:** Chair of Romance Linguistics (French-Italian), Institute of Romance Studies, TU Dresden

Prof. Dr. Anna-Maria De Cesare, Dr. Stefan Koch, M.A. Tom Weidensdorfer,  
M.A. Michela Gargiulo, M.A. Claudia Rausch

Automated texts belong to a rapidly evolving field and have a high degree of disruptive power. From a technical standpoint, they are generated with new and increasingly sophisticated techniques, involving the use of algorithms and (soft) artificial intelligence. From a communicative perspective, they are in the process of revolutionizing the way texts are conceived, produced, distributed, and consumed. As they grow in number, improve in quality, and expand in variety, automated texts are set to become an important part of our lives over the next decade.

The 1<sup>st</sup> Ai-ROM Conference aims at bringing together researchers from all areas of Romance Linguistics and neighboring fields (Communication Science, Media and Journalism Studies, Computational Linguistics) to reflect on automated texts from a variety of perspectives: theoretical, descriptive, and practical. We understand ‘automated texts’ in a broad sense, including formats as diverse as texts generated by AI-powered smart agents, informally called “chatbots”, producing written and/or oral outputs (e.g., ChatGPT, as well as “virtual assistants” such as Alexa, Siri etc.), template-based automated texts (A.I. Anchor, Gabriele, Tobi) automated neural machine translations (produced by DeepL, Google Translate etc.), and texts generated by writing assistants (e.g., rytr.com).

We welcome abstracts on single Romance languages and varieties, on comparisons between Romance languages or between Romance and Germanic languages. The conference is open to all theoretical and methodological approaches. Special interest lies in corpus-based and corpus-driven analyses as well as qualitative and quantitative analyses. Papers presented at the conference will be published in a special issue of an international journal (TBA).

### Topics of interest for the Ai-ROM conference include but are not limited to:

- Taxonomies of automated texts, reflections on category boundaries and hybrid forms
- Comparisons between automated and human language (written, oral, hybrid)
- Linguistic features of automated texts (i.e., lexical, morphological, syntactic)
- Graphic properties of automated texts (e.g., punctuation, emoji, emoticons)
- Linguistic expression of stereotypes as well as gender, race, age, and other forms of biases
- Textual properties of automated texts (nature of textual units, cohesive markers, thematic progression, implicit communication, authorship etc.)
- Text automation and information packaging
- Pragmatic and discursive properties of automated texts
- Challenges and opportunities of text automation for teaching and learning practices
- Challenges and opportunities of automated texts for smaller Romance varieties

### Abstract submission and notification of acceptance

If you are interested in participating in the conference, **please submit your abstract** (in English or a Romance language: Italian, French, Spanish, Portuguese) **by April 15, 2023** to [ai.rom@mailbox.tu-dresden.de](mailto:ai.rom@mailbox.tu-dresden.de). The abstract must be anonymous and include between 400 and 500 words (references excluded). It must provide information on the language(s) considered, the phenomenon analyzed, the data used and the method of investigation. It should also indicate some (provisional) results.

The Scientific Committee will **announce acceptance / rejection of proposals by April 30, 2023**.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/369087672>

# Dialog chatbot as an interactive online tool in enhancing ESP vocabulary learning

Article · March 2023

DOI: 10.1108/SJLS-10-2022-0072

---

CITATIONS  
0

READS  
5

5 authors, including:



Fawaz Qasem

University of bisha

21 PUBLICATIONS 72 CITATIONS

[SEE PROFILE](#)



Ahmed A. Al Khateeb

King Faisal University

17 PUBLICATIONS 121 CITATIONS

[SEE PROFILE](#)

# Dialog chatbot as an interactive online tool in enhancing ESP vocabulary learning

Chatbot as a tool for ESP vocabulary learning

Fawaz Qasem

*Department of English, College of Sciences and Arts, and the Applied College at Al-Namas, University of Bisha, Bisha, Saudi Arabia*

Mukhtar Ghaleb

*Department of Information Systems, College of Computing and Information Technology, University of Bisha, Al Namas, Saudi Arabia and Faculty of Computer Science and Information Technology, Sana'a University, Sana'a, Yemen*

Hassan Saleh Mahdi

*Department of English, Hodeidah University, Hodeidah, Yemen and English Language Centre, Taif University, Taif, Saudi Arabia*

Ahmed Al Khateeb

*Department of English Language, College of Arts, King Faisal University, Al-Ahsa, Saudi Arabia, and*

Hind Al Fadda

*Department of Curriculum and Instructions, King Saud University, Riyadh, Saudi Arabia*

Received 5 October 2022  
Revised 18 November 2022  
5 January 2023  
29 January 2023  
Accepted 1 February 2023

## Abstract

**Purpose** – Based on an experimental study on English for Specific Purposes (ESP) students, at the Business Department at the University of Bisha, the purpose of the study is to examine the effect of chatbot use on learning ESP in online classrooms during COVID-19 and find out how Dialogflow chabot can be a useful and interactive online platform to help ESP learners in learning vocabulary well.

**Design/methodology/approach** – The research paper is based on an experimental study of two groups, an experiential group and a controlled group. Two tests were carried out. Pre-tests and post-test of vocabulary knowledge were conducted for both groups to explore the usefulness of using the Dialogflow chatbot in learning ESP vocabulary. A designed chatbot content was prepared and included all the vocabulary details related to words' synonyms and a brief explanation of words' meanings. An informal interview is another tool used in the study.

---

© Fawaz Qasem, Mukhtar Ghaleb, Hassan Saleh Mahdi, Ahmed Al Khateeb and Hind Al Fadda. Published in *Saudi Journal of Language Studies*. Published by Emerald Publishing Limited. This article is published under the Creative Commons Attribution (CC BY 4.0) licence. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial and non-commercial purposes), subject to full attribution to the original publication and authors. The full terms of this licence may be seen at <http://creativecommons.org/licenses/by/4.0/legalcode>

The authors would like to thank the two anonymous reviewers for their constructive comments on the paper.

*Competing interests:* The authors declare no competing interests.

*Ethical approval:* The authors obtained approval from the deanship of scientific research at their affiliated university. xxxxxxxx. Further approvals were obtained from the invited universities before the participants were invited to participate in the study.

*Informed consent:* The participants were informed during the data collection process that the participation was voluntary; all the information was treated with confidentiality.



The purpose of using the interview with the participants was to elicit more data from the participants about using the chatbot and about how and in what aspects chatbot using the conversational program was useful and productive.

**Findings** – The findings of the study explored that the use of chatbots plays a major role in enhancing and learning ESP vocabulary. That was clear as the results showed that the students who used the chatbot Dialogflow in the experimental group outperformed their counterparts in the control group.

**Research limitations/implications** – The study displays an important pedagogical implication as the use of chatbots could be applied in several settings to improve language learning in general or learning ESP courses in particular. Chatbot creates an interesting environment to foster build good interactions where negotiation of meaning takes place clearly seems to be of great benefit to help learners advance in their L2 lexical development.

**Originality/value** – Examining and exploring whether the use of chatbots plays a major role in enhancing and learning ESP vocabulary in English as Foreign Language setting.

**Keywords** Chatbot, English for specific purposes (ESP), ESP vocabulary, English as second language (ESL), Online platform

**Paper type** Research paper

## 1. Introduction

Teachers always look for new pedagogical strategies and techniques to help their students learn better, faster and more in-depth. We live in a time when education is more readily available than it has ever been, with the help of fast technology development. The integration of technology and learning languages has become fruitful and a fact we cannot escape. Adult learners are often time-constrained and making time for learning is rarely a priority. Chatbots can assist your current students and are not limited by human constraints such as forgetting and recall. New conversational learning technologies (chatbots) can simulate a conversation with a colleague when it comes to learning and training. A training event should appear and feel like a natural conversation between you and a co-worker, so it can be very personal, to the point and enjoyable. A chatbot is a conversational agent (an artificial intelligence [AI] program) that communicates with users using natural language and makes decisions based on predefined rules.

Using chatbots in language teaching and learning has been examined in several studies (e.g. [Laurillard, 2002](#), [Smutny and Schreiberova, 2020](#)). The COVID-19 pandemic breakout made all institutions look for various and replacing options and recent academic approaches and strategies to engage students in the learning process and to create good interactive environments for students and teachers. There are a few studies to examine the effect of chatbots on learning English for specific purposes (ESP) in online classrooms during COVID-19 or as promising tool to help students use while learning or communicate with their teachers. Thus, the current study aims at examining how Dialogflow chatbots can be common online platforms to help ESP learners learn vocabulary well.

## 2. Literature review

### 2.1 *ESP and digital and technology age*

Technology with its fast-digital development has a strong and influential role in ESP. It helps to a great extent in creating good, rich and real-life environments to build various interesting ESP projects and produce and design tailor-made curriculums. The need for English for specific situations leads to the spread and the importance of ESP in many disciplines and environments and many ESP projects have been designed within the framework of needs analysis. The emergence of technology gave more strength to ESP with its educational nature that is based on courses designing and providing authentic and natural real life and real-life situations. Many research works have shown how technology, Computer-Assisted Language Learning contributed greatly to the area of ESP in terms of teaching/training, designing and developing or helping ESP learners.

The integration of technology and ESP was productive and the application of technology in the field of ESP ([Wang, 2015](#)). [Dashtestani and Nadezda \(2015\)](#) similarly recommended that

“ESP teachers make attempts to use a wide range of technologies in their ESP courses in order to maximise student participation and engagement in language learning”. Technology with its nature where ESP teachers and practitioners can bring a good and interactive environment, plays important role in the ESP learners’ engagement and participation, especially in the case of using a Course Management System in the ESP discipline ([Maulan and Ibrahim, 2012](#)). In modern various contexts where technology is used, learners become more independent and can have good space to interact with their teachers and students.

## *2.2 Artificial intelligence, chatbot and language learning*

The development of information communication technologies has received unprecedented growth including the expansion in the applications of AI. AI is a vibrant field that intersects with other major flourishing domains such as machine learning, deep learning and cognitive computing ([Bini, 2018](#)). The evolution of AI has accompanied the rise of numerous web tools, neural networks and virtual applications ([Battineni et al., 2019; Jadhav and Thorat, 2020](#)). Therefore, there is evidence that AI systems are changing the nature of education and the process of language learning ([Bii, 2013](#)). The variability of AI definitions in the literature is obvious since it has become an integral part of numerous fields including education and language learning. For instance, AI refers to computational programs that help in simulating and mimicking human intelligence, such as problem-solving and learning ([Shouval et al., 2020](#)). AI creates naturalistic conversational interactions directed towards comprehensive second and foreign language learning ([Divekar et al., 2021](#)). Furthermore, AI deals with the study and design of algorithms that perform tasks or behaviours that a person could improve to require intelligence if a human were to do it. [Riedl \(2019\)](#) has asserted that AI involves intelligent systems, known as intelligent agents, which are also accountable for taking decisions on their own as they could achieve comparable actions to humans such as Alexa, Cortana or Google Assistant. Several AI-designed applications have been identified to support language learners such as interactive conversational tools (i.e. chatbots), three-dimensional face automatic recognition assistants and translation technologies ([Blyth, 2018](#)). To be specific, according to [Haristiani \(2019\)](#), a chatbot is a computer program based on AI that can carry out conversations through audio or text which has possibilities for extending language learning.

In conjunction with what is stated earlier, the chatbot is a computer program that simulates human conversation through voice commands or text chats or both. Chatbot, short for chatterbot, is an AI feature that can be embedded and used through any major messaging application. [Fei and Petrina \(2013\)](#) also define a chatbot program as a distinctive program from other computer applications that are built on mimicking intelligent conversation as human users via deploying auditory or textual procedures. Chatbots are known as advanced forms of human–machine interaction with automatic conversational agents which link the users and machines with the assistance of natural language processing ([Luo et al., 2019](#)). Considering this term, experts in this field use chatbots to refer to robotic actions since it is a special kind of robot that is designed to stimulate conversation with human users via the Internet ([Kim, 2018](#)).

Vocabulary is an essential part of learning a language and to think, learn and express about the world. Expanding the knowledge of words provides unlimited access to new information; particularly in a second or foreign language. There are numerous methods for learning vocabulary, but memorisation is one of the most common that is often practiced in rote learning ([Yang and Dai, 2011](#)). In this sense, [Chen et al. \(2020\)](#) contended that the memorisation of words is inseparable from the context of vocabulary learning. According to [Muhammad et al. \(2020\)](#), Dialogflow is a platform for natural language understanding that facilitates the design and integration of conversational user interfaces into mobile applications, web applications, devices, bots, interactive voice response systems, etc.

### 3. Methodology

#### 3.1 Research questions

The research article addresses the following questions.

*RQ1.* Is there any significant difference between the learners who used chatbots and the learners who used the traditional approach in learning Business English?

*RQ2.* What is the learners' perception of using chatbots in learning business English terms?

*RQ3.* What are the advantages of using chatbots in learning ESP courses, such as Business English?

#### 3.2 Participants

Two classes at the University of Bisha, Saudi Arabia, were selected to participate in the study. The participants were undergraduate students who were doing their BA program in Business English at the university. Research ethics were maintained throughout this research. The researcher obtained approval letters from the ethics research committee at the University of Bisha numbered (UB-18-2020). The participants enrolled in Business English as one of the courses of the program Business Administration. The course aimed to help them be equipped with English to help them at specific, professional and academic levels. Arabic was the first language of all the participants. All of them were male students. Then, the two classes were randomly assigned to treatment groups, one that practiced chatbots and one that practiced a traditional approach. The participants were 20 in the experimental group and 20 in the control group. The first group was the experimental group which taught the course for 12 weeks with a help of chatbot dialogue. The second group was the control group which was taught English without the support of chatbots. The participants of the two groups had the same level of English proficiency.

#### 3.3 Materials and instrument

Since the study focus was the use and learning of vocabulary by ESP learners of Business English, the materials manipulated in the study include 10 units selected from the common book of [Mascull \(2010\)](#) that is entitled "Business Vocabulary in Use Advanced with Answers". The topics selected were related to the majority of the participants, for instance, meetings, negotiations, career ladder, etc. The designed chatbot content included all the vocabulary details related to words' synonyms and brief explanations of words' meanings.

#### 3.4 Data collection

Since this research is based on an experimental study of two groups, experiential and controlled groups, pre-tests of vocabulary knowledge were conducted for both groups. The sample size of the participants was the same ( $n = 20$ ) in each group. The participants in the study were given the same time and academic level of input. Two researchers conducted the experiment. The meetings with the students were twice a week for the two groups. The participants in the experimental group only were asked to use chatbots during classroom activities and tasks or when they have their outside assignments outside the classrooms. In order to investigate the influence of chatbot use, post-tests were conducted for both groups after 12 weeks.

*3.4.1 Data collection tools.* 3.4.1.1 The test. The instrument used to collect the data was a vocabulary test. The vocabulary test was created by the authors based on the textbook taught to the students ([Mascull, 2010](#)). The test was designed to examine how participants used chatbots to learn new terms in Business English. The Cronbach's alpha was 0.80, which is considered good. The test was made up of 20 items. The pre-test consisted of the same terms that were used in the post-test but in different contexts. The participants were asked to

choose the correct option. Each sentence with the correct option was given one point. Thus, the total points for the test were 20 points.

3.4.1.2 The interview. An interview is one of the tools used to get more data from the participants and to explore more data on the research questions. The purpose of using the interview with the participants was to elicit more data about how and in what aspects using a chatbot conversational program was useful and productive. An interview protocol question was designed and distributed to some participants. For the convenience of the study and participants, the questions of the interviews were written and sent to the participants via Google form. The questions include some hints on the advantages of using the chatbot conversational program. Ten students participated in this interview.

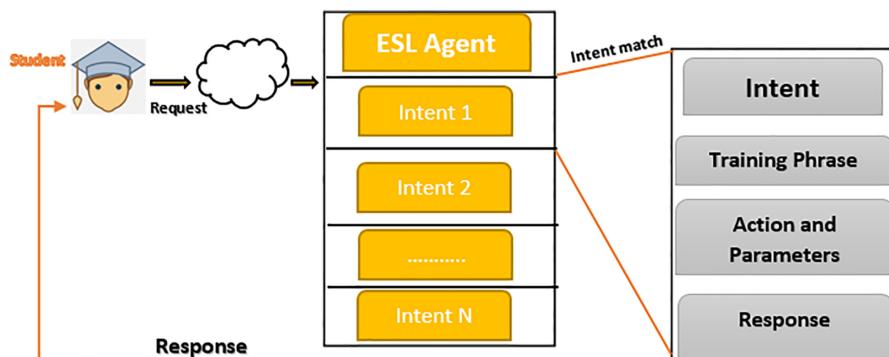
3.4.1.3 Chatbot. The chatbot is a computer program that is available on mobiles as an application and the use of chatbots is becoming easy and interaction can happen anywhere as they are accessible. Participants in this study use their mobiles and check ESP words in the designed chatbot dialogue flow. [Figure 1](#) shows how the chatbot works.

ESL Agent is a program that works on dialogflow on the Google platform, where a group of intent is written on a specific topic. After that, when the student checks the meaning of a specific word, the program searches in the previously entered intent group, and when it finds any match, it responds to the inquiry. Training phrases are examples of what the end user can say, knowing that it is not necessary to identify all possible examples because integrated machine learning extends with other relevant phrases.

### 3.5 Procedures

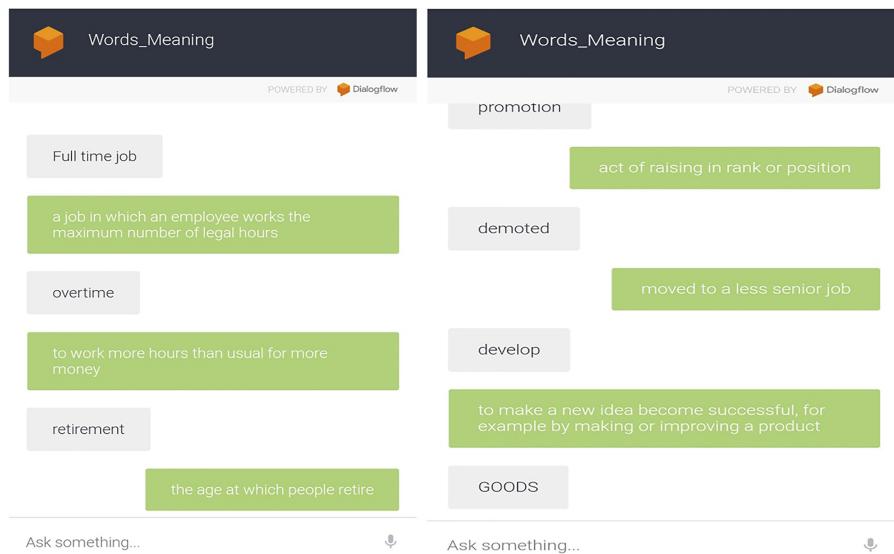
Prior to commencing the experiment, the participants took the pre-test to ensure that they were homogenous in terms of their proficiency level. After that, the participants were assigned to the experimental and control groups and attended 12 sessions in the academic year 2020/2021. In the first session, the participants in the two groups took the pretest. In addition, the participants in the experimental group which used chatbots were provided with the necessary information regarding how the software works. Then, the two groups received the instruction for 12 sessions. In the experimental group, the participants received instructions on how to use a chatbot. For each session, the participants were required to use chatbots in the inside classroom and outside classroom activities and tasks. For instance, check the examples (see [Figure 2](#)):

In the control group, the same instruction was practiced through the traditional approach. The participants were taught the same content but without using the chatbot. In the last session, the participants in the two groups received the post-test to check and determine how much the participants benefited from the chatbot over the term of applying and checking the ESP vocabulary in the chatbot.



**Figure 1.**  
English as second language (ESL) agent architecture

---



**Figure 2.**  
Students' samples  
of ESP vocabulary  
meaning search  
in Chatbot

### 3.6 Data analysis

In order to analyse the data, Statistical Package for the Social Sciences version 22 was used. First descriptive statistics including mean and standard deviation were calculated for each group. Then an independent-sample *t*-test was employed for comparing the two groups' performance. In order to examine changes in learners' performance from the pre-test to the post-test, a paired-samples *t*-test was used. To explore the learners' perceptions of using the chatbot in learning business English terms, a thematic content analysis was done for open-ended questions and a descriptive analysis was performed for close-ended questions.

## 4. Results

To answer the first research question, descriptive statistics and *t*-tests were performed. The first research question of the study attempted to explore whether using the chatbot had any significant impact on learning Business English terms. To analyse the data, first descriptive statistics were calculated. Table 1 presents the results.

A *t*-test analysis of the pre-test was used to check whether there was a significant difference between the two groups before the experiment. This analysis indicated that there

Groups	<i>N</i>	Mean	Std. deviation	Std. error mean	Levene's test for equality of variances		<i>t</i> -test for equality of means				
					<i>F</i>	Sig.	<i>t</i>	Df	Sig. (2-Tailed)	Mean difference	$\eta^2$
Chatbot group	20	9.05	3.74833	0.83815	0.000	1.000	-0.240	38	0.812	-0.30000	0.039
Control group	20	9.35	4.14570	0.92701							

**Table 1.**  
Descriptive statistics  
of the students'  
performance of pre-test

was no significant difference in the level of the two groups ( $p = 0.812$ ). The magnitude of the differences in the means (mean difference =  $-0.300$ ) was small ( $\eta^2 = 0.039$ ).

To answer the first research question of whether the participants' learning business English was improved as a result of the intervention of using a chatbot, the mean and standard deviation across posts showed variation in the participants' performance. Results are presented in [Table 2](#).

As shown in [Table 2](#), there was a statistically significant difference between the mean score of the experimental and control group after being treated with the instructions;  $t (38) = 2.90$ ,  $p = 0.006$ , two-tailed. The magnitude of the differences in the means (mean difference =  $3.150$ ) was large ( $\eta^2 = 0.426$ ).

The second research question was about the learners' perceptions of using chatbots in learning business English terms. Qualitative and quantitative analyses of the interview questions were performed. The qualitative analysis was done using thematic content analysis for open-ended questions. The quantitative analysis was done using descriptive analysis for close-ended questions.

Regarding their answers to the first question of whether they found chatbots easy or difficult. All of them stated that it was easy to be used. The second question was about if they thought that chatbots can be used in other courses. In total, 80% agreed that chatbots could be used in other courses, and 20% felt that they might be used. The third question was about using chatbots in other courses. All the respondents stated that they used chatbots in other courses. The fourth question asked the respondents to determine whether the chatbot helped them to remember words better. In total, 80% of them stated that chatbot helped them remember words better and 20% of the respondents stated that chatbot might be the reason for remembering words. The fifth question was about using chatbots for their future academic career. In this case, 80% of the respondents stated that they would use a chatbot in their future academic career, and 20% of the respondents stated that they might use chatbot in their future academic career.

The second part of the interview contained open-ended questions. The first question in this part was about the benefits of learning Business English terms. The respondents stated that they recognised the meanings of the terms, and it helped them so much to get the meaning of the words quickly. In addition, chatbots explained words that have different meanings in an easy and organised way. The second question was about the faults of chatbots. They stated that chatbots were nice tools to be used and only a few faults that had been noticed. First, some terms were not included in the chatbot. Second, chatbots need an Internet connection. The third question was about their suggestions to improve the chatbot. They suggested that more words should be added. Also, they suggested chatbots can be designed as an application and used anywhere when there is an Internet connection. They urged us to use chatbots in all other courses.

Groups	N	Mean	Std. deviation	Std. error mean	t-test for equality of means				
					T	Df	Sig. (2-Tailed)	Mean difference	$\eta^2$
Chatbot group	20	14.85	3.63137	0.81200	2.90	38	0.006	3.150	0.426
Control group	20	11.70	3.21346	0.71855					

**Note(s):** \* $p < 0.05$

**Table 2.**  
Descriptive statistics of  
the students'  
performance of the  
post-test

## 5. Discussion

Adapting technology in learning L2 in general, or ESP field, is increasing. The purpose of the current study was to examine the impact of implementing a chatbot in learning ESP vocabulary (Business English vocabulary). With respect to the impact of implementing chatbot in learning Business English, the analysis of the data indicated that the experimental group outperformed their counterparts in the control group. The finding of this study is in line with several previous studies and recent studies. Recently, for instance, [Bailey and Almusharraf \(2021\)](#) examined how productive using a chatbot is in learning L2. They checked the incorporation of a digital storytelling chatbot system and investigated how students' perceptions of the story bot interactions. The study found that story bots helped the students to meet their goals to learn L2 and increased their participation. In a similar vein, [Wollny et al. \(2021\)](#) recently explored the positive use of chatbots as a promising tool in education in terms of skill development, education efficiency, learners' motivation and education availability.

The understanding of the ESP vocabulary well and being developed with the help of chatbot approved that the integration of digital and technological software and applications can act as a gateway for better understanding ESP vocabulary and language skills and improving ESP materials. For instance, [Butler-Pascoe \(2011\)](#) gave importance to technology integration with ESP for better course design and development. Using chatbots as a tool was positive in acquiring ESP vocabulary and this supports teachers to use various digital and online tools in improving learners' skills in L2 or ESP vocabulary and courses' contents. Similarly, many studies approved that ESP learning is enhanced by the use of technology. For, instance, the use of wikis in ESP instruction was positive and learners were active in learning ESP patterns ([Felea and Stanca, 2014](#)). Moreover, with their communicative nature like chatbots, using blogs has been productive and useful in learning ESP textbooks and developing ESP knowledge. In the same way, using blogs was useful and positive in learning ESP especially in developing the learners' classroom communication and in improving learners' autonomous and independent learning ([Chong, 2010](#)). Similarly, a designed model for developing a chatbot was assessed and it was found that the chatbot was useful as an extra tool to carry out academic and administrative tasks and facilitate communication between students and academic staff ([Mendoza et al., 2022](#)).

Most of the interview responses of the experimental group were supportive towards the use of chatbots. Most of the participants suggest that chatbot was useful, and they find it a good tool to help them engage and learn ESP English vocabulary. They found it easy to use as the chatbot was directed and focused on ESP Business English words and no challenges or difficulties were noticed while practicing and learning ESP vocabulary. The responses showed that chatbots and any technological application integration can be beneficial and such a study can be a starting point for more focused fieldwork studies to explore the effectiveness of technology integration in ESP and in educational pedagogy in general. Most of the participants were with idea of teaching all ESP courses and other courses in the future with the use of some applications, programs and softwares as chatbots. This study supports the fact of the upcoming smart tools and software with high and advanced technology in relation to AI that would lead to a new revolution and trend in learning and future academic research and online applications such as chatGPT ([O'Connor, 2022](#)).

## 6. Conclusion

The research discussed the implementation of chatbots in learning ESP vocabulary. It was found that using ESP vocabulary within the technology environment (chatbot) had an increasing influence on ESP learners. Performance results of the ESP vocabulary in the pre-test and post-test revealed that the experimental group significantly outperformed the control group in learning ESP words in the post-test. The study shows clearly that the use of chatbots

acts well in enhancing and learning ESP vocabulary. This suggests that ESP teachers should make use of chatbots applications and other digital and distance technology in teaching ESP vocabulary and in engaging ESP students in learning better. Using chatbots offers several opportunities for language learners as well as teachers. Therefore, the study displays some suggestions that can be applied to improve language learning using chatbots. First, interaction tasks where negotiation of meaning takes place clearly seem to be of great benefit to help learners advance in their L2 lexical development. Chatbot creates an interesting environment to foster such interactions. Based on the theory of noticing and attention, the study suggests that cognitive factors such as attention and depth of processing are the key elements to be used to facilitate L2 vocabulary development through synchronous interactive tasks using chatbots. Further studies with a larger sample in EFL and ESL contexts would be useful to highlight and assess the positive pedagogical implications of chatbot use in enhancing vocabulary learning and the skills of English.

## References

- Bailey, D. and Almusharraf, N. (2021), "Investigating the effect of chatbot-to-user questions and directives on student participation", *2021 1st International Conference on Artificial Intelligence and Data Analytics, CAIDA 2021*, May, 85-90, doi: [10.1109/CAIDA51941.2021.9425208](https://doi.org/10.1109/CAIDA51941.2021.9425208).
- Battineni, G., Canio, M.D., Chintalapudi, N., Amenta, F. and Nittari, G. (2019), "Development of physical training smartphone application to maintain fitness levels in seafarers", *International Maritime Health*, Vol. 70 No. 3, pp. 180-186, doi: [10.5603/IMH.2019.0028](https://doi.org/10.5603/IMH.2019.0028).
- Bii, P. (2013), "Chatbot technology: a possible means of unlocking student potential to learn how to learn", *Educational Research*, Vol. 4 No. 2, pp. 218-221, available at: <http://psych.athabascau.ca/html/chatterbot/ChatAgent>
- Bini, S.A. (2018), "Artificial intelligence, machine learning, deep learning, and cognitive computing: what do these terms mean and how will they impact health care?", *The Journal of Arthroplasty*, Vol. 33 No. 8, pp. 2358-2361.
- Blyth, C. (2018), "Immersive technologies and language learning", *Foreign Language Annals*, Vol. 51 No. 1, pp. 225-232.
- Butler-Pascoe, M.E. (2011), "The history of CALL: the intertwining paths of technology and second/foreign language teaching", *International Journal of Computer-Assisted Language Learning and Teaching (IJCALLT)*, Vol. 1 No. 1, pp. 16-32.
- Chen, H.-L., Vicki Widarso, G. and Sutrisno, H. (2020), "A chatbot for learning Chinese: learning achievement and technology acceptance", *Journal of Educational Computing Research*, Vol. 58 No. 6, pp. 1161-1189.
- Chong, E.K.M. (2010), "Using blogging to enhance the initiation of students into academic research", *Computers and Education*, Vol. 55 No. 2, pp. 798-807.
- Dashtestani, R. and Nadezda, S. (2015), "The use of technology in English for specific purposes (ESP) instruction: a literature review", *The Journal of Teaching English for Specific and Academic Purposes*, Vol. 3 January, pp. 435-456, available at: [file:///C:/Users/dell/Desktop/Fawaz ref/304-1148-1-PB.pdf](file:///C:/Users/dell/Desktop/Fawaz%20ref/304-1148-1-PB.pdf)
- Divekar\*, R.R., Drozdal\*, J., Chabot\*, S., Zhou, Y., Su, H., Chen, Y., Zhu, H., Hendler, J.A. and Braasch, J. (2021), "Foreign language acquisition via artificial intelligence and extended reality: design and evaluation", *Computer Assisted Language Learning*, February, doi: [10.1080/09588221.2021.1879162](https://doi.org/10.1080/09588221.2021.1879162).
- Fei, Y. and Petrina, S. (2013), "Using learning analytics to understand the design of an intelligent language tutor – chatbot lucy", *International Journal of Advanced Computer Science and Applications*, Vol. 4 No. 11, pp. 124-131, doi: [10.14569/ijacsa.2013.041117](https://doi.org/10.14569/ijacsa.2013.041117).
- Felea, C. and Stanca, L. (2014), "Wiki tools in teaching English for specific (academic) purposes - improving students' participation", *Lecture Notes in Computer Science (Including Subseries*

- Haristiani, N. (2019), "Artificial intelligence (AI) chatbot as language learning medium: an inquiry", *Journal of Physics: Conference Series*, Vol. 1387 No. 1, doi: 10.1088/1742-6596/1387/1/012020.
- Jadhav, K.P. and Thorat, S.A. (2020), "Towards designing conversational agent systems", *Computing in Engineering and Technology*, Springer, pp. 533-542.
- Kim, N. (2018), "Chatbots and Korean EFL students' English vocabulary learning", *Journal of Digital Convergence*, Vol. 16 No. 2, pp. 1-7, available at: file:///C/Users/dell/Desktop/Fawaz\_ref/ChatbotsandKoreanEFLStudentsEnglishVocabularyLearning.pdf
- Laurillard, D. (2002), *Rethinking University Teaching: A Conversational Framework for the Effective Use of Learning Technologies*, Routledge, London.
- Luo, X., Tong, S., Fang, Z. and Qu, Z. (2019), "Frontiers: machines vs humans: the impact of artificial intelligence chatbot disclosure on customer purchases", *Marketing Science*, Vol. 38 No. 6, pp. 937-947, doi: 10.1287/mksc.2019.1192.
- Mascull, B. (2010), *Business Vocabulary in Use: Intermediate with Answers and CD-ROM*, Cambridge University Press, Cambridge.
- Maulan, S.B. and Ibrahim, R. (2012), "The teaching and learning of English for academic purposes in blended environment", *Procedia - Social and Behavioral Sciences*, Vol. 67, pp. 561-570, doi: 10.1016/j.sbspro.2012.11.361.
- Mendoza, S., Sánchez-Adame, L.M., Urquiza-Yllescas, J.F., González-Beltrán, B.A. and Decouchant, D. (2022), "A model to develop chatbots for assisting the teaching and learning process", *Sensors*, Vol. 22 No. 15, p. 5532.
- Muhammad, A.F., Susanto, D., Alimudin, A., Adila, F., Assidiqi, M.H. and Nabhan, S. (2020), "Developing English conversation chatbot using dialogflow", *2020 International Electronics Symposium (IES)*, pp. 468-475.
- O'Connor, S. (2022), "Open artificial intelligence platforms in nursing education: tools for academic progress or abuse?", *Nurse Education in Practice*, Vol. 66, 103537.
- Riedl, M.O. (2019), "Human-centered artificial intelligence and machine learning", *Human Behavior and Emerging Technologies*, Vol. 1 No. 1, pp. 33-36, doi: 10.1002/hbe2.117.
- Shouval, R., Fein, J., Savani, B., Mohty, M. and Galski, H. (2020), "Machine learning and artificial intelligence in haematology", *British Journal of Haematology*, Vol. 192, doi: 10.1111/bjh.16915.
- Smutny, P. and Schreiberova, P. (2020), "Chatbots for learning: a review of educational chatbots for the Facebook Messenger", *Computers and Education*, Vol. 151 February, doi: 10.1016/j.compedu.2020.103862.
- Wang, Y.-C. (2015), "Promoting collaborative writing through wikis: a new approach for advancing innovative and active learning in an ESP context", *Computer Assisted Language Learning*, Vol. 28 No. 6, pp. 499-512.
- Wollny, S., Schneider, J., Di Mitri, D., Weidlich, J., Rittberger, M. and Drachsler, H. (2021), "Are we there yet? A systematic literature review on chatbots in education", *Frontiers in Artificial Intelligence*, Vol. 4, pp. 1-18, doi: 10.3389/frai.2021.654924.
- Yang, W. and Dai, W. (2011), "Rote memorization of vocabulary and vocabulary development", *English Language Teaching*, Vol. 4 No. 4, pp. 61-64, doi: 10.5539/elt.v4n4p61.

### About the authors

Fawaz Qasem is currently working as Assistant Professor of Applied Linguistics, at the Department of English, University of Bisha. He has published and presented many research papers and attended various international conferences, workshops and webinars. He works as an editor and reviewer in various international journals. His research interests include Linguistics, Applied Linguistics and Acquisition of L2, Psycholinguistics, Sociolinguistics, Corpus Linguistics, educational technology and ESP. Fawaz Qasem is the corresponding author and can be contacted at: [fqaqasem@ub.edu.sa](mailto:fqaqasem@ub.edu.sa)

Mukhtar Ghaleb is Assistant Professor of Computer Networks Department at the University of Bisha, Saudi Arabia. He received his B.S. degree in Computer Information systems from Zarka Private University, Jordan, in 2004. He received his M.S. degree in networking and distributed computation from University Putra Malaysia (UPM), Malaysia, in 2008. He began his pursuit of his career with an appointment at Sana'a University, Yemen. He received his Ph.D. degree in computer networks from UPM, in 2014. Currently, his research interests are mobile data gathering, routing protocols, power consumption, performance modelling and simulation, Terrestrial and underwater Sensor Networks, AI and sentimental analysis.

Hassan Saleh Mahdi is Assistant Professor of applied linguistics in the Department of English, University of Bisha, Saudi Arabia. His research interests are computer-assisted language learning (CALL), Mobile-assisted language learning (MALL) and second language vocabulary acquisition. He has published several articles related to these topics in leading journals such as Journal of Educational Computing Research, Journal of Computing in Higher Education and Journal of Psycholinguistic Research. He reviewed many manuscripts for journals with high-impact factors in language learning such as *ReCALL Journal* and *Language Teaching Research Journal*.

Ahmed Al Khateeb is Associate Professor at the English Language Department at King Faisal University, Saudi Arabia. He holds a PhD in Applied Linguistics and Modern Languages from the University of Southampton in the UK. He is a winner of a Fulbright scholarship and visiting scholar at the University of Massachusetts, Amherst. His research interests include technology-enhanced language learning (TELL), advanced learning technologies, telecollaboration and language learning, intercultural communication and psychology of language learners and their cognitive behaviours.

Hind Al Fadda is Associate Professor in College of Education at King Saud University. Her field of specialist is teaching English as a second language (TESOL) and mainly using technology in teaching (CALL). She had many published many studies in her field and she also contributed to many conferences in second language teaching and in education in general.

---

For instructions on how to order reprints of this article, please visit our website:

[www.emeraldgroupublishing.com/licensing/reprints.htm](http://www.emeraldgroupublishing.com/licensing/reprints.htm)

Or contact us for further details: [permissions@emeraldinsight.com](mailto:permissions@emeraldinsight.com)

## Research Article

# Constructing a Data-Driven Model of English Language Teaching with a Multidimensional Corpus

Dongyan Chen  <sup>1,2</sup>

<sup>1</sup>College of International Studies, Beibu Gulf University, Guangxi 535015, China

<sup>2</sup>Academy of Language Studies, University of Technology MARA, Negeri Selangor 40450, Malaysia

Correspondence should be addressed to Dongyan Chen; chendongyan1120@163.com

Received 21 February 2022; Accepted 28 March 2022; Published 28 June 2022

Academic Editor: Gengxin Sun

Copyright © 2022 Dongyan Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this study, a multidimensional corpus English teaching model is constructed using the data-driven model. This study uses a data-driven collection of massive amounts of data to generate a multidimensional corpus. The data-driven generation of a multidimensional corpus to build a teaching model is studied, and the principle of data driven, the computational process, and the characteristics of the corpus are analyzed. Due to the deficiency of data-driven modeling without correlating process variables with quality variables, this study adopts an artificial intelligence algorithm and analyzes the basic principle, computational process, and advantages and disadvantages of the method. The model is simulated and verified for multidimensional corpus and English teaching. To address the shortcomings of the AI algorithm, which has a complex computation process and no orthogonal decomposition of the data space, the autoregressive latent structure projection algorithm is designed by integrating the autoregressive idea with the artificial intelligence (AI) algorithm. This algorithm can orthogonally decompose the sample data space and simplify the modeling process. Finally, the algorithm is validated by simulation. To verify the results of the teaching model, the fuzzy C-means clustering algorithm is combined with the autoregressive latent structure projection algorithm in this study. The sample data used in the modeling are divided into categories, and the affiliation function is calculated for each category. The affiliation function is used to calculate the affiliation of the online calculation results for each category, and the final evaluation results are obtained based on the fuzzy comprehensive evaluation method. Finally, taking junior students as an example, the simulation is carried out to verify the effectiveness of the English teaching model. The research results show that the corpus-based English flipped classroom teaching model improves English teaching methods, enhances students' English proficiency and independent learning ability, and provides a practical basis for English teaching model exploration.

## 1. Introduction

Technology continues to penetrate the field of education, and students interact with various platforms, generating a large amount of learning behavior and achievement data, which have significant educational value when accumulated over time [1]. When teachers teach and research, they should actively explore and use student data to diagnose student problems and improve teaching methods so that teachers can transform their teaching and research and teaching from empirical and process oriented to scientific and personalized. The application of education data in education can promote the development of education informatization and education modernization. At present, data analysis

platforms are gradually being built in primary and secondary schools, and educational data are being accumulated. Teachers can use student data to gain a deeper understanding of students' learning needs, gain insight into the path of learners' learning behavior to improve teaching, enhance teaching effectiveness, and promote teachers' professional development [2]. Data-driven teaching has been the frontier of international education information development, and data-driven teaching has four characteristics: scientific, precise, intelligent, and personalized. Data-driven teaching and research are the links before data-driven teaching is carried out, teaching and research are the foundation of teaching, and scientific teaching and research are conducive to improving the teaching effect. Although

some schools have not built data platforms, with the increasing awareness of data use, teachers should have an in-depth understanding of the data-driven teaching and research process in advance from the principle of understanding the process of data generation, acquisition, processing, and analysis. At present, the theoretical research on multidimensional corpus English teaching and research in China are insufficient [3]. There is not enough theoretical research on multidimensional corpus ELT teaching and research in China to guide teachers in corpus teaching and research practice. The purpose of this study is to study the data-driven corpus teaching and research process, to develop a corpus English teaching and research program, and to guide teachers step by step to utilize the corpus of student data so that student data can help teachers in their teaching decisions.

As the national demand for high-end foreign language talents continues to increase, foreign language education is facing higher and higher requirements for talent cultivation, and the cultivation of international composite talents with “one specialization and multiple abilities” and “one proficiency and multiple skills” has become one of the important directions of foreign language education reform in universities [4]. In the process of cultivating complex talents, academic English teaching plays an important role, because academic English highlights the instrumental characteristics of English, which can meet the practical needs of students’ professional study and cultivate students’ ability to use English for work and scientific research. In recent years, the process of national education informatization has continued to advance, and information technology has revolutionized higher education, especially foreign language education, triggering deep changes in foreign language education philosophy, teaching organization, and teaching methods [5]. The Guide to Teaching English at University requires teachers to build and use microcourses and catechisms, use online high-quality educational resources to transform and expand teaching contents, and implement hybrid teaching modes such as flipped classes based on classroom and online courses [6]. Thus, this study tries to construct a corpus-based English flipped classroom teaching model, empirically test its teaching effect, and provide a practical reference for English teaching model exploration.

## 2. Related Works

Data-driven teaching and research abroad originated at the beginning of the 21st century and were earlier called professional learning communities (PLCs) and later called data teams (DTs). WestEd is a nonprofit research, development, and service organization dedicated to improving the education of children, youth, and adults. Led by Ellen Mandinach, senior research scientist and director of decision data, WestEd researchers believe that teachers should not be taught data literacy skills in isolation; they believe that data literacy skills should be taught in conjunction with data use processes [7]. Jamal et al. also conducted an empirical study of the impact of data team implementation on student achievement by implementing the Harvard Data Wisdom

Improvement Process at Leisure Elementary School, led by the school’s leadership, to train the school’s teachers on how to organize collaborative work, lead teachers to dig deeper into student data to identify instructional problems, and then create instructional solutions for students and teachers based on the team’s findings [8]. After implementation, Leisure Elementary’s test scores reached their highest level since 2009, bridging the gap between the test scores of special education students and the general student population. In the direction of teacher data literacy research, various experts and scholars offered their insights on data literacy. Zhang and Han believe that teacher data literacy consists of three major components: data awareness, data competence, and data ethics [9]. Di Gangi used the ACTS academic quality evaluation report form to develop new teaching strategies by first reading the data, identifying problems, focusing on them, analyzing the causes, addressing them, and exploring the teaching in three steps so that the average score of this class changed from below the regional average to above the average, which verified the effectiveness of data-driven teaching research and highlighted the application value of student achievement data. This demonstrates the effectiveness of data-driven teaching and research and highlights the value of student achievement data [10]. According to the law of large numbers, when the number of training samples tends to be infinite, the empirical risk approaches the expected risk, and the prediction can be accurate for new samples.

European linguists and educators proposed the use of corpora as an aid in the teaching and learning process of foreign languages. The corpus is a very important branch of corpus linguistics that can be used as a teaching aid in foreign language teaching and is considered an effective teaching method that encompasses two main aspects: one is to directly teach corpus knowledge, using the corpus as a means and resource for language teaching; the other is to indirectly use it as a tool for lexicography, grammar reference, grammar teaching, and other multimedia courseware and as a corpus and computer-based language learning software and testing tool. Tsai, a linguist, made an important contribution to corpus applications, arguing that vocabulary teaching is the primary task of foreign language teaching [11]. At the beginning of the 21st century, Tsai again advocated the use of corpus-based chunking in foreign language teaching. He suggests that learners independently learn through authentic corpora so that students can understand the meaning and usage of vocabulary. This is a typical example of the application of the corpus-based block teaching method in language teaching, emphasizing the authenticity of the corpus and the scientific nature of the computer as a supplementary teaching tool in language teaching [12]. Piotrkowicz et al. analyzed the use of chunks in Chinese English learners’ language output using the “corpus of Chinese English learners’ spoken language” and showed the quantity and quality of chunks used by Chinese students in both languages’ organization and content selection that are not satisfactory [13]. Hooshyar et al. argue that the corpus is more applicable to higher education and that the richer and faster updating of the corpus in the case

of postgraduate academic learning proves that the corpus can better help students in international learning and communication [14].

### 3. Construction of a Data-Driven Multidimensional Corpus-Based ELT Model

**3.1. Data-Driven Model Design.** As we all know, the data-driven algorithm is different from a model-driven algorithm; in that it no longer needs to build a physical model for a specific problem but uses the data generated by the problem for various tasks such as monitoring, evaluation, and control. Artificial intelligence (AI), as the mainstream algorithm of data-driven technology, has made breakthroughs in four aspects, such as algorithm, data, computing power, and framework, in the past two decades, and thus has been widely used in various fields [15]. Artificial intelligence algorithms can be mainly classified into three categories: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is the most used type of AI algorithm, usually using a training set consisting of  $n$  “input-output” pairs  $\{(x_i, y_i)\}_{i=1}^n$  to continuously “learn” the mapping relationship between input and output  $y = f(x)$ . Supervised learning is essentially the process of “fitting” the mapping relationship, where the optimization parameters in the mapping are continuously adjusted by an optimization algorithm to reduce the “loss” value of the training samples, also known as empirical risk, i.e.,

$$\begin{aligned} \text{Loss}(f) &\leq f(x) \sum e^x, \\ y_i &= t_j x_i + \frac{x_j}{t_{i+1} + t_j}. \end{aligned} \quad (1)$$

According to the law of large numbers, when the training samples tend to infinity, the empirical risk tends to be closer to the expected risk, and thus, the prediction can be accurate for new samples. Common supervised learning algorithms include shallow common algorithms such as decision trees (DTs), support-vector machines (SVMs), and neural networks (NNs), and convolutional neural networks (CNN), recurrent neural network (RNN), and other deep neural networks. In this study, deep neural networks are mainly used, but ordinary neural networks are also used for some applications where the mapping relationship is relatively simple.

To build an AI model that meets the practical application requirements using the above algorithms, we need to realize the tedious and underlying code development such as data storage, model building, optimization training, and hardware acceleration. It is impractical to go through such a tedious development process for each model building. Therefore, open-source or commercial software frameworks have been developed to address this problem, which can be used by developers in related fields to accelerate the process of model building, data access, and training inference. Software frameworks include both hardware acceleration of software frameworks developed by hardware vendors for deep learning and software frameworks that focus on model building and fast training for developers.

The principal component analysis is often applied to multimetric performance evaluation, which can solve the problem of the high complexity of analysis and evaluation caused by the excessive number of metrics in the process of multimetric evaluation. Due to the strong correlation of indicators in the original data, the information reflected by indicators will overlap. The principal component analysis (PCA) algorithm uses as few independent new indicators as possible for the original sample data and reflects the process information carried by the original sample data as much as possible, through which the correlation between the indicators of the original sample data is eliminated and the dimensionality of the original sample data is reduced. The principle of the algorithm is shown in

$$X = [x_1, x_2, \dots, x_n] = \begin{Bmatrix} x_1 & \dots & x_n \\ x_{11} & \dots & x_{1n} \\ x_{n1} & \dots & x_{nn} \end{Bmatrix}, \quad (2)$$

$$\text{STD}_{ij} = S_i + \frac{X_j}{x_{ij}}. \quad (3)$$

The current breakthrough in data-driven technology is due to the improvement of the “source of intelligence” algorithm, rather than the accumulation of massive data and the improvement of computing power, which are the most important factors for breakthroughs in the field of artificial intelligence in the past decade. Data are the core of data-driven algorithms, and the scale and quality of the data directly affect the accuracy and generalization ability of the trained models. To carry out the task of transient stability assessment using massive data samples, researchers have tried various data-driven algorithms in recent years to continuously improve the accuracy and computational efficiency of transient stability assessment. In general, there are two main types of ideas for constructing stability assessment models with the help of data-driven algorithms: one is to construct stability boundaries with the help of massive data, based on which the stability conclusions are drawn by judging the relative position of the current operating state and the boundaries. It is true that some algorithms skip the step of boundary construction and directly draw stability conclusions based on some existing data samples in the relative neighborhood. Another class of ideas is to construct mapping relations from system measurement information or system operation and disturbance characteristics to stability conclusions with the help of massive data samples. Although the constructed mapping relations are like stability boundaries in mathematical essence, they have significant differences in the way of thinking and are, therefore, considered as another class.

To enhance the feature extraction performance of the feature extractor, its order and parameters must be carefully designed with the help of convolutional, activation, and pooling layers. In general, the deeper the neural network

structure, the higher the accuracy of the model on more complex tasks, but it also leads to longer training time, poorer convergence characteristics of optimization, and more severe overfitting. Therefore, the essence of deep neural network model design is a trade-off between model complexity and accuracy. In common classical models, convolutional and activation layers are usually combined to extract features from the input. It is obviously unrealistic to go through such a tedious development process for each model building. To exactly determine how many convolution and activation layers are needed, these two layers can be added to the model until the accuracy of the model no longer significantly improves. At the same time, pooling layers can be added to the model to reduce the training parameters without significantly sacrificing the accuracy of the model. Based on this principle, the convolutional neural network feature extractor section shown in Figure 1 is designed and used in the example analysis section of this study: it contains five convolutional layers, five activation layers, and three pooling layers. The critical line's transfer power must not exceed the available transfer capability (ATC), and the feature extractor is used to discover more implicit "rules" to distinguish the stability of each sample to be evaluated.

After extracting the input feature information, the network structure of the fully connected layer is constructed to establish the mapping between the features extracted from the training samples and their stability findings to predict the stability findings of the new samples. Most deep learning models, including AlexNet and VGG, use a network structure with 3 layers of fully connected layers. Considering the complex and high-dimensional nonlinear nature of the transient stability problem, the number of layers is set to 3 here. In addition, considering that the fully connected layer is prone to overfitting during training, the dropout technique is used here to enhance the robustness of the network by forcing some hidden neurons to zero.

**3.2. Building a Multidimensional Corpus ELT Model.** A corpus (plural corpora), as its name implies, is a storehouse of linguistic materials, a database of written and spoken language stored in a computer for research purposes. The distinguishing feature of a corpus is that the language materials are real materials in actual use, covering a wide range of fields such as literature, business, and educational translation, and the number of words covered is in the hundreds of millions. The corpus is preferred by scholars in various fields because of the observability and verifiability of the data it provides, its shareability, and its ease of retrieval [16]. Corpora use random sampling methods to collect naturally occurring continuous language according to certain linguistic rules, through texts or language fragments to build a large electronic textbase with a certain capacity. Currently, corpus linguistics is widely used in foreign language research. The research in foreign language teaching is divided into two types. On the one hand, there are applied studies that use the materials in the corpus as research texts. This type of research mainly uses learner corpora, such as the Chinese Learner English Corpus (CLEC) and the

International Corpus of Learner English (ICLE), to conduct a comprehensive study of various lexical or grammatical error features in the writing or speaking of English learners at home and abroad. To determine how many convolution and activation layers are needed, you can keep adding these two layers to the model until the accuracy of the model no longer significantly improves. At the same time, pooling layers can be added to this model to reduce training parameters without significantly sacrificing model accuracy.

Microlearning is a product of the deep integration of information technology and education teaching and has become an important teaching resource. With the characteristics of "prominent theme, short and concise, interesting, and wide application," microlessons help share high-quality teaching resources and make learning possible anytime and anywhere. To meet the fragmented learning needs of students, we have developed a series of microlessons on academic English vocabulary, grammar, reading, translation, writing, and listening, guided by the principles of integration of academic and interest, unification of thematic and application, and coordination of focus and relevance. Based on the corpus-based academic English teaching platform, we have built a flipped English classroom teaching model, as shown in Figure 2, which aims to support the corpus-based teaching platform and combines the advantages of traditional classrooms to interconnect online interactive teaching and offline interactive teaching to form a flipped classroom organism, so that the time and space of English teaching and learning can be infinitely extended and the purpose of improving students' English ability and academic literacy can be achieved.

The basic idea of partial least squares is to decompose the process variable data space into two subspaces according to the magnitude of correlation with the quality variable  $X$ , i.e., the subspace  $X'$  containing the correlation between the process variable  $X$  and  $Y$ , and the residual matrix  $X_0$ , which is uncorrelated with the quality variable  $Y$ . However, the nonlinear iterative partial least squares (NIPALS) algorithm used in this algorithm has difficulty in ensuring that  $X'$  and  $X_0$  are mutually orthogonal, and the algorithm loops once to obtain a score vector, leading to high computational complexity. YIN et al. proposed an autoregressive projection to latent structure by combining autoregressive ideas with partial least squares. The algorithm uses historical data, establishes the corresponding regression coefficient matrix, and performs orthogonal decomposition of the sample space of historical data of process variables based on the principle of the magnitude of the correlation between process variables and quality variables. The algorithm can solve the problem of the high complexity of standard partial least squares operation. With the complete decomposition of the quality variable data space  $Y$ , the matrix can specifically reflect the correlation between the process variable  $X$  and the quality variable  $Y$ , which is defined as the regression coefficient matrix according to the form of the regression algorithm.  $e'$  denotes the residual space of the quality variable data space  $Y$ . With a complete decomposition of the quality variable data space,  $E'$  should be independent of the process variable data space  $X$ .

Input   Data normalization      Feature extractor      Classifier      Output

FIGURE 1: Structure of the feasible convolutional neural network.

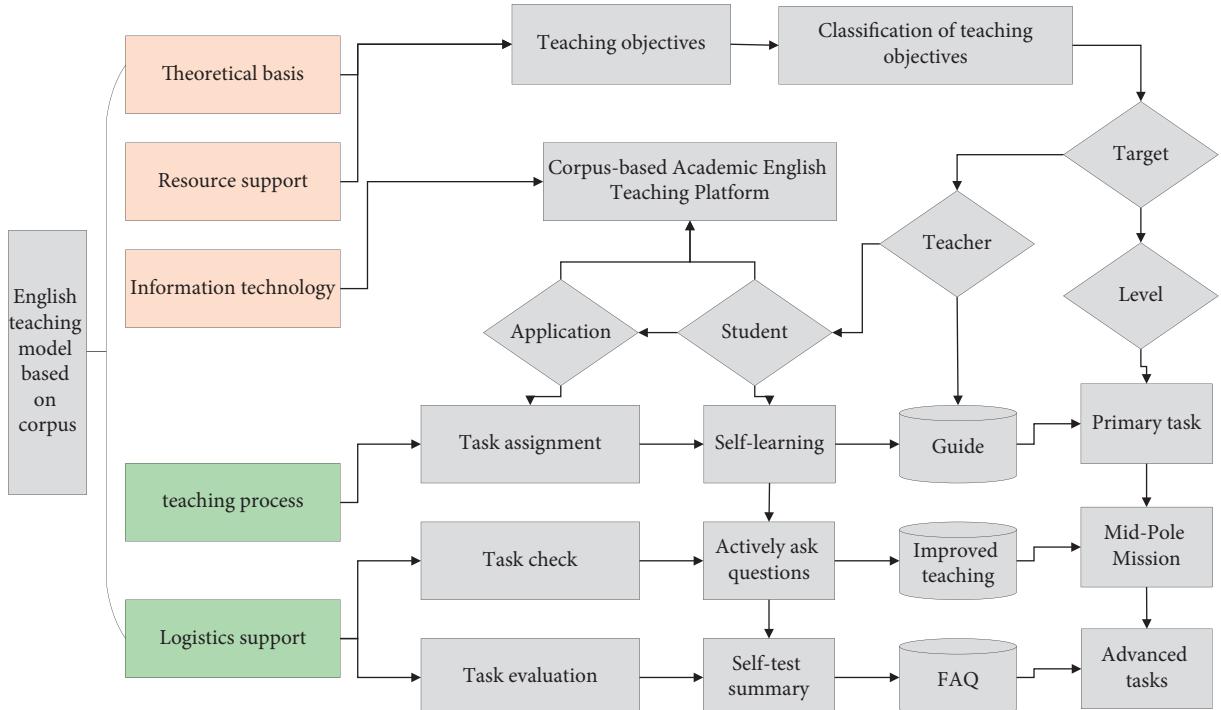


FIGURE 2: English teaching model based on the corpus.

$$\begin{aligned}
 E(e_y, x) &= \text{cov}(e_y x^T) \leq 1, \\
 X &= X' + \hat{X} \\
 &= P_M P_t^M - X' P' P_M^t.
 \end{aligned} \tag{4}$$

From Figure 3, it is easy to find that when the number of latent variables  $p > 2$ , the modeling process is more stable because the autoregressive latent structure projection (AR-PLS) algorithm does not use the nonlinear iterative partial least squares (NIPALS) method used in regression modeling. In addition, the standard partial least squares method requires a given number of latent variables for modeling, and the determination of the number of latent variables has a significant impact on the process monitoring. The calculation process of the algorithm is simple, easy to understand, and has high efficiency in calculating high-dimensional

sample data. The fuzzy C-means algorithm converts the traditional fuzzy clustering method into an optimization problem with a constraint function. There is no theoretical method to determine the number of latent variables, and more practical methods such as cross-validation methods are used to determine the number of latent variables, which brings uncertainty to the established industrial process monitoring models. In contrast, the autoregressive latent structure projection (AR-PLS) algorithm no longer requires the number of latent variables to be set, which has certain advantages, as shown in Figure 3.

To clarify the research progress of data-driven teaching and research models, foreign teaching and research models were sorted out. At present, foreign research on data-driven teaching and research models is more mature, and foreign countries call data-driven teaching and research models as collaborative data team procedure, datawise process, etc.

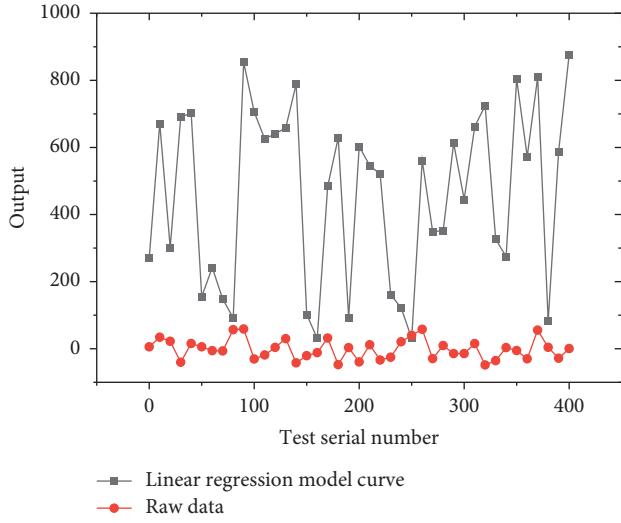


FIGURE 3: Autoregressive latent structure projection model performance test results.

Kim Schildkamp et al. are dedicated to studying how schools use student data and how to organize and build data teams so that they can better serve teaching and learning. Mandinach developed a complete approach to data use, data literacy for teachers (DFLT), and a conceptual framework for teachers. As shown in Figure 4, content knowledge, curriculum knowledge, learners' knowledge and learner characteristics, awareness of educational purposes and values, general pedagogical knowledge, pedagogical content knowledge, and educational background knowledge are used as inputs for the data use of the teaching process.

In the traditional machine learning approach, model modeling requires the provision of three datasets: training, validation, and testing. These are applied to model fitting, model hyperparameter tuning, and assessing model generalization capabilities, respectively. Of course, it is the best choice if the required experimental information can be collected in the real physical network, but data collection in the real network faces problems such as the impact of real physical network information collection on the performance of the present network, the high cost of real physical network information collection, and the topology of the real physical network topology in the case of guaranteed homogeneity with a single topology [6]. However, the nonlinear iterative partial least squares (NIPALS) algorithm used in this algorithm is difficult to ensure that  $X'$  and  $X_0$  are mutually orthogonal, and to obtain a score vector, the algorithm loops once, resulting in high computational complexity. Therefore, the network simulation service hopes to construct the network simulation dataset required for modeling the network delay performance algorithm in the network delay inference service through a discrete event-driven network simulator.

## 4. Results Analysis

**4.1. Data-Driven Model Results.** The network simulation dataset should be able to meet the modeling requirements of

the network delay performance algorithm in the network delay inference service. On the other hand, the network managers in the inferred system management platform need to follow the standards defined in the network simulation service for a specific network scenario, and only through the inferred system management platform can the network managers complete the use of network delay performance prediction and historical information query for a specific network scenario [17]. The network simulation dataset built by the network simulation service runs through the entire development and uses the process of the data-driven network QoS inference system, which puts higher requirements on the reliability of the network simulation dataset.

Cluster analysis is an important research element in the field of data mining, and cluster analysis algorithms are widely used in many fields of daily life. Clustering analysis can explore the structural features inside the data, classify and label the generated data, and then uncover the potential and unknown information inside the data. Cluster analysis is a classical unsupervised classification method that uses mathematical methods to automatically classify a sample set of data without giving classification principles in advance. The fuzzy C-means cluster analysis algorithm can solve the requirement of fuzziness, which is difficult to solve by hard cluster analysis methods and is a coarse division of sample data. The algorithm has a simple and easy-to-understand computational process and has high efficiency in computing high-dimensional sample data. The fuzzy C-means algorithm converts the traditional fuzzy clustering method into solving optimization problems with constraint functions. As shown in Figure 5, the use of the higher-order data-driven arbitrary polynomial chaos expansion method implies the use of the higher-order polynomial basis functions in polynomial approximation, and more polynomial basis functions will bring closer approximation to the simulation results.

The performance evaluation based on the autoregressive latent structure projection algorithm proposed in this study first uses the production process prediction model established by the autoregressive latent structure projection algorithm, then uses the fuzzy C-means cluster analysis algorithm to calculate the affiliation of the output variable data in the modeled data for each performance level, and obtains the affiliation function of each variable for each performance level [11]. In the performance evaluation of online data, due to the lag in the production process, the output variables are first predicted using the prediction model and the input variable data, and then, the affiliation function is used to calculate the affiliation of each output variable predicted value for each performance level, and finally, the performance level belonging to that moment is obtained using the fuzzy operator to provide a reference for the field operators. It is not difficult to find from Figure 3 that when the number of latent variables  $p > 2$ , since the autoregressive latent structure projection (AR-PLS) algorithm does not use the partial least squares method, the nonlinear iterative partial least squares (NIPALS) method used in regression modeling makes the modeling process more stable.

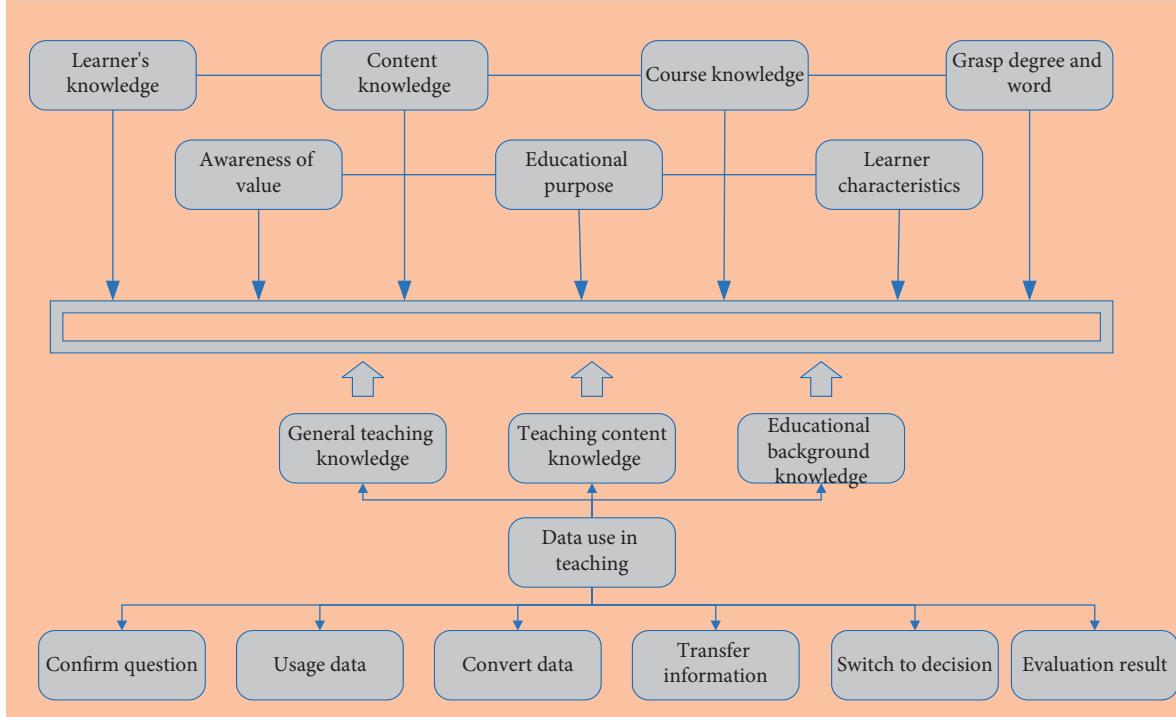


FIGURE 4: Data literacy of teachers' conceptual framework.

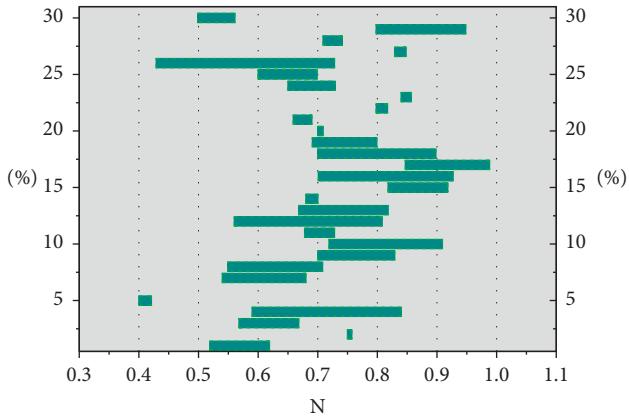


FIGURE 5: Eigenvalue method to analyze the number of clusters.

The ecological foreign language teaching model in the computer network environment is a theoretical teaching procedure that uses the theory of foreign language teaching and educational ecology as the common support theory, the optimal combination of all teaching elements as the construction base, the computer network technology as the development intermediary, the use of different teaching strategies to maximize the presentation of teaching content, to achieve the set teaching goals, the structural framework of various teaching activities, and the collection of teaching methods. It is the direction of rational construction and optimization of modern foreign language teaching mode. This model advocates open teaching information selection according to the teaching needs and mostly carries out task-based teaching activities in the form of collaboration and

mutual assistance between teachers and students, which helps to cultivate students' language communication and comprehensive application skills. Because of the current development of the integration of computer network technology and foreign language courses, we must comprehensively compare and analyze the ideal one under the premise of considering various factors such as hardware and software conditions of computer network technology, teaching objectives of foreign language courses, teaching staff's willingness to choose information and network teaching literacy, students' learning motivation, level and network application ability, and auxiliary background management mechanism of network teaching. To build a foreign language teaching model that meets its development conditions has clear development goals, and wide development space, a reasonable series of planning and design can be carried out. The performance test results of the data-driven model are shown in Figure 6.

In this chapter, a framework of data-driven transient stability boundary generation and an online stability evaluation algorithm are proposed. To this end, a critical transient stability sample sampling and resampling mechanism is first proposed to accelerate the generation of sufficient critical transient samples in the high information entropy region near the stability boundary to provide a data basis for transient stability boundary generation. In addition, a critical operation and perturbation scenario screening mechanism is developed to further reduce the search space of the system, which provides a feasible solution to the challenges of "dimensional disaster" and "combinatorial explosion" faced by the stable boundary construction problem. Overall, the algorithm not only significantly

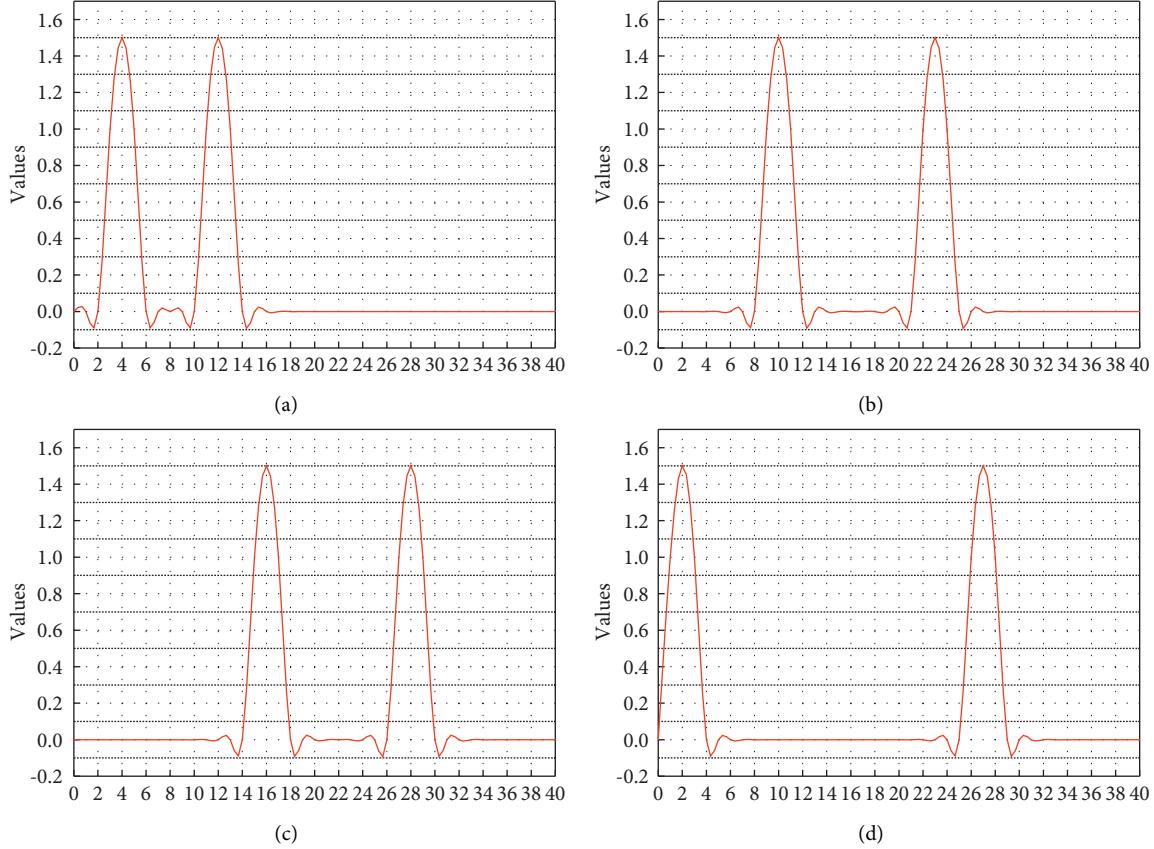


FIGURE 6: Data-driven model performance test results.

improves the efficiency of key transient sample generation and accelerates the speed of transient stable boundary generation but also significantly reduces the computational burden with the designed running point tracking and periodic boundary update mechanism, which makes it possible to update the state of the multidimensional corpus in real time based on data driven. In addition, the standard partial least squares method requires a given number of latent variables in modeling, and the determination of the number of latent variables has a great impact on process monitoring.

**4.2. Simulation Experiments of Multidimensional Corpus English Teaching Model.** The new computer network teaching environment requires teachers to change their teaching concepts, adhere to the “student-centered” teaching concept, act as an analyst of students’ learning needs, a guide of learning direction, and a supervisor of overall activities, and assist students to realize the transformation of the central subject position of classroom teaching activities and the improvement of independent learning ability on the internet [18]. The students’ learning needs are analyzed, learning directions are guided, and overall activities are supervised. However, research statistics show that only 47.6% and 44.3% of teachers and students, on average, believe that teachers can act as directional guides in multimedia classrooms and students’ online learning, respectively, which shows that more than half of teachers have

almost neglected their leading role. If they cannot design classroom activities, they cannot provide students with opportunities for active thinking and feedback, and they cannot effectively motivate students to learn. At the same time, as many as 43.6% and 47.3% of students are evaluated as marginalized in multimedia classroom teaching activities and passive recipients of knowledge in online independent learning, resulting in a serious lack of their role as learning subjects. Their personal learning needs are not clear, their ability to make independent decisions is poor, and the learning process is only blindly and passively accepted. There are even 34.7% of students who are evaluated as internet information losers, and their learning effect is even worse.

For this study, we selected English III students from the university of X to implement this data-driven teaching and research project. University of X is a second-level undergraduate institution with a strong overall student body and a strong faculty and has sufficient capacity to complete the experiment. As shown in Table 1, five teachers participated in this school, one as the subject team leader, one information system management teacher, and two subject teachers. The information system management teacher did not participate in the entire data-driven teaching and learning activity; the teacher participated in the data-driven teaching and learning in stage 3 by providing support for the teaching and learning data survey. Student data support was needed when conducting student problem queries. Querying student data was not easy, the school did not have all student

TABLE 1: Composition of teachers in the experimental group.

Participant	Function	Teaching age	Subject	Requirements or not
Director Wang	Subject group leader	10	English	Yes
Teacher Ma	Teacher	6	English	Yes
Ms. Zhao	Teacher	6	Science	Yes
Mr. Xu	Teacher	7	English	Yes
Instructor Sun	Data coach	—	Computer science	Yes

data recorded in an electronic system, and access to the system administrator revealed that the school had a data system, but only stored midterm and final grades and text-based teacher evaluations of students' each semester. This study led teachers to record weekly paper test scores in electronic form so that teachers could use the data for visual analysis. If the school had more comprehensive data, it would support teachers in making more accurate decisions, and the lack of easy access to data seriously affects the effectiveness of data-driven teaching and research.

First, the data coach explained the types of teaching data for teachers, and then, the data coach instructed the three teachers and the data system management teacher to recall and query the data stored in this grade in our school to record the data of this data-driven teaching and research class. This data record form records the data of this grade level in seven aspects: data number, data name, data generation time, data form, data storage location, public object, and data use. The four main types of data related to teaching and learning in grade 4 are unit test scores, student classroom performance ratings, student grouping data, formal routine assessments, and formal classroom assessments. From the unit test scores, we can diagnose the weak points of students' knowledge, and by correlating student grouping data, student classroom performance scores, and unit test scores, we can explore the degree of correlation between students' usual performance and grades. In short, these data are the basis for our data-driven teaching and research.

Before the experiment began, the author administered a controlled output vocabulary test in both the experimental and control classes, and after the test, the test scores were entered into SPSS 22.0 for descriptive statistics data analysis; 50 test papers were valid. The lowest score in the experimental class was 14, the highest score was 36, and the overall mean score in the experimental class was 23.98. While the lowest score in the control class was 12, the highest score was 33, the overall mean score was 22.98, and the difference between the overall mean scores of the two classes was small. The data analyzed by independent samples t-test, the result of chi-square test, the value of F statistic is 0.004; therefore, the variance of the two classes' performance is chi-square. T-test results should be selected, which equal variances assumed (assuming equal variances), with the first line of data as the test result. T-statistic is 0.286, degrees of freedom is 98, and thus above the 0.05 significance level, the null hypothesis is accepted as valid, that is, there is no significant difference between the test results of the experimental and control classes. From the mean of the two classes' scores and the independent samples t-test, there is no significant difference between the scores of the experimental class and the control

class before the experiment, which can be compared and analyzed. The statistical results of the experimental test are shown in Figure 7.

According to Figure 7, it can be learned that there is very little difference in the number of students using the language blocks in the two classes before the experiment. Comparing the data of the post-test, the number of language blocks in the experimental class was significantly higher than that in the control class: the experimental class used a total of 431 language blocks, with 10.3 blocks per capita, while the control class used a total of 397 language blocks, with 9.7 blocks per capita. In other words, the number of blocks used in the experimental class significantly increased in the post-test, which had a positive effect on the writing performance. According to linguist Ding Yanren's study, it was confirmed that students' writing performance was related to the number of blocks used. Therefore, it can be inferred that the more the number of blocks used by the learners in writing, the higher the writing scores. Thus, it is known that the number of chunks used has a direct effect on writing performance, which means that increasing students' meaningful input of chunks has a positive effect on improving writing performance. Cluster analysis can mine the internal structural features of the data, classify and mark the generated data, and then discover the potential and unknown information in the data.

The model generalization capability was verified, and model strengths and weaknesses were compared on the 17 node network simulation test dataset provided by the NSS subsystem, in which the model version number model9389 had good model generalization capability on the test dataset provided by the NSS subsystem. Meanwhile, by comparing the performance of the model versions model9389 and model9450 on the test set, we can conclude that the log-log loss function is more applicable than the MSE mean square error loss function in the framework of the algorithm model for network transmission delay performance evaluation in this thesis. In the requirement analysis for QISMP inference system management, the tests on personalization management, account management, model version management, model prediction, and history information query, respectively, show that QISMP encapsulates various types of service interfaces to provide network managers with the ability to perform network latency performance evaluation and model version management for special network scenarios. The performance test results of the multidimensional corpus teaching model are shown in Figure 8.

Based on the analysis of the requirements for the simulation packet generation, routing and forwarding, and node

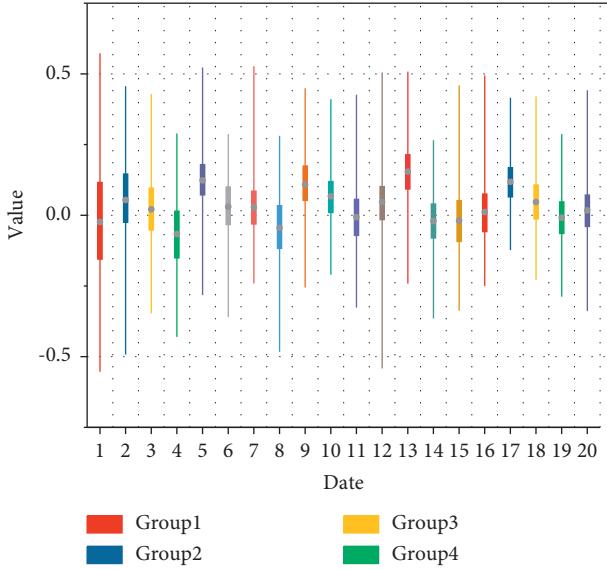


FIGURE 7: Four groups of control classes' tests with more statistical results.

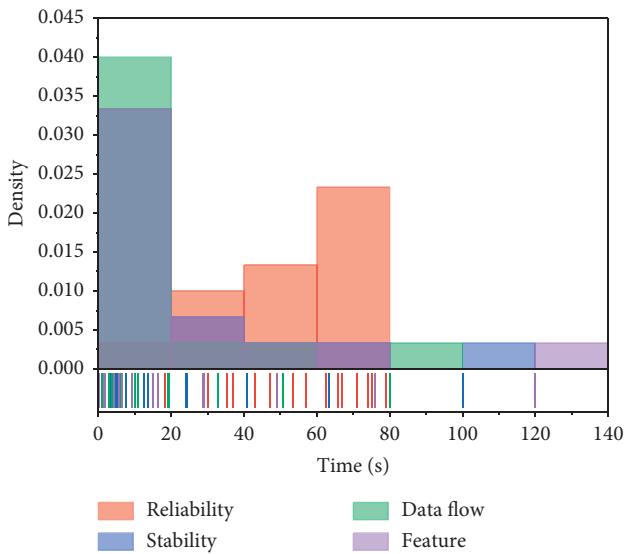


FIGURE 8: Performance results of the multidimensional corpus English teaching model.

packet management functions that the NSS subsystem network simulation service should have, the test results of the simulation network basic functions under a 14 node network topology with simplified configuration parameters show that the NSS subsystem can complete the transmission performance simulation of specific network scenarios based on the discrete event-driven network simulator OMNeT++. This study provides network simulation data support for the network delay performance modeling by deep learning methods in the NDIS subsystem. The data-driven network QoS inference system based on this thesis is divided into the NSS subsystem, NDIS subsystem, and QISMP. Through the functional tests of the three subsystems, respectively, in this chapter, it can be concluded that the NSS subsystem based

on the event-driven network simulator OMNeT++ can effectively provide data support for the network latency performance evaluation model modeling in the NDIS subsystem and the NDIS subsystem. The network delay performance evaluation model NMBGNN designed and implemented in the NDIS subsystem can provide reliable delay prediction service for QISMP, and QISMP can provide model prediction and model version management for network managers with a reasonably designed platform functions.

## 5. Conclusions

In this study, we constructed a data-driven multidimensional corpus-based English teaching model and formed a data-driven teaching and research model. By implementing data-driven teaching and research in university and continuously revising the process and model in practice, we finally built a relatively perfect process model of data-driven teaching and research activities. A framework of transient stable fast batch assessment algorithm is proposed, and the cascaded convolutional neural network is constructed to adaptively select the simulation time window for the samples to be assessed and terminate the time-domain simulation as early as possible while ensuring the accuracy of the assessment conclusion, which is achieved to reduce the overall computational burden of the batch assessment task. In response to the deficiencies in the examination of the relationship between output and input variables, this study explains the basic principles, modeling steps, and advantages and disadvantages of the partial least squares method, introduces the autoregressive latent structure projection algorithm to address the deficiencies of the partial least squares method in the modeling process, and analyzes the modeling principles, steps, and characteristics of the algorithm. In English teaching, teachers are beginning to pay attention to students' sense of experience and to gradually return the classroom center to students. Teachers are also changing their teaching methods and teaching tools to improve teaching standards, following the concept of integrating modern smart teaching technology with English curriculum teaching, further deepening the discussion of combining smart classroom tools with independent learning English skills, and providing new ideas and ways to effectively improve the development of English teaching quality. It provides new ideas and ways to effectively improve the quality of English teaching.

## Data Availability

The data used to support the findings of this study and acknowledgment reference [1] are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work was supported by the 2021 Guangxi Higher Education Undergraduate Teaching Reform Project “Research and Practice on Construction of Ideological and Political Evaluation Index System of College English Curriculum Based on CIPP” (project no.: 2021JGB271).

## References

- [1] R. Yan, G. Geng, Q. Jiang, and Y. Li, “Fast transient stability batch Assessment using cascaded convolutional neural networks,” *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2802–2813, 2019.
- [2] S. C. Silva, T. C. Ferreira, and R. M. S. Ramos, “Data-driven and psycholinguistics-motivated approaches to hate speech detection,” *Computación Y Sistemas*, vol. 24, no. 3, pp. 1179–1188, 2020.
- [3] M. Mussetta and A. Vartialatis, “Writing across the curriculum in ELT training courses: a proposal using data-driven learning in disciplinary assignments,” *International Journal of Teaching and Learning in Higher Education*, vol. 30, no. 2, pp. 300–307, 2018.
- [4] I. Ivaska and S. Bernardini, “Constrained language use in Finnish: a corpus-driven approach,” *Nordic Journal of Linguistics*, vol. 43, no. 1, pp. 33–57, 2020.
- [5] D. Hooshyar, M. Yousefi, and H. Lim, “A systematic review of data-driven approaches in player modeling of educational games,” *Artificial Intelligence Review*, vol. 52, no. 3, pp. 1997–2017, 2019.
- [6] B. C. Runck, S. Manson, E. Shook, M. Gini, and N. Jordan, “Using word embeddings to generate data-driven human agent decision-making from natural language,” *Geo-Informatica*, vol. 23, no. 2, pp. 221–242, 2019.
- [7] N. I. Khursanov, “On the theoretical and practical foundations of language corpora,” *Asian Journal of Multidimensional Research*, vol. 10, no. 9, pp. 311–318, 2021.
- [8] J. Jamal, A. Shafqat, and E. Afzal, “Teachers’ perceptions of incorporation of corpus-based approach in English language teaching classrooms in Karachi, Pakistan,” *Liberal Arts and Social Sciences International Journal (LASSIJ)*, vol. 5, no. 1, pp. 611–629, 2021.
- [9] C. Zhang and J. Han, “Multidimensional mining of massive text data,” *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 11, no. 2, pp. 1–198, 2019.
- [10] M. A. Di Gangi, G. Lo Bosco, and G. Pilato, “Effectiveness of data-driven induction of semantic spaces and traditional classifiers for sarcasm detection,” *Natural Language Engineering*, vol. 25, no. 2, pp. 257–285, 2019.
- [11] K.-J. Tsai, “Corpora and dictionaries as learning aids: inductive versus deductive approaches to constructing vocabulary knowledge,” *Computer Assisted Language Learning*, vol. 32, no. 8, pp. 805–826, 2019.
- [12] A. Akbari, “Translation quality research,” *Babel. Revue internationale de la traduction/International Journal of Translation*, vol. 64, no. 4, pp. 548–578, 2018.
- [13] A. Piotrkowicz, K. Wang, J. Hallam, and V. Dimitrova, “Data-driven exploration of engagement with workplace-based assessment in the clinical skills domain,” *International Journal of Artificial Intelligence in Education*, vol. 31, no. 4, pp. 1022–1052, 2021.
- [14] D. Hooshyar, M. Yousefi, and H. Lim, “Data-driven approaches to game player modeling,” *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–19, 2018.
- [15] S. Crossley and M. M. Louwerse, “Multi-dimensional register classification using bigrams,” *International Journal of Corpus Linguistics*, vol. 12, no. 4, pp. 453–478, 2007.
- [16] A. Batliner, S. Steidl, and C. Hacker, “Private emotions versus social interaction: a data-driven approach towards analysing emotion in speech,” *User Modeling and User-Adapted Interaction*, vol. 18, no. 1, pp. 175–206, 2008.
- [17] L. Flowerdew, “Applying corpus linguistics to pedagogy,” *International Journal of Corpus Linguistics*, vol. 14, no. 3, pp. 393–417, 2009.
- [18] A. A. M. Al-Gamal and E. A. M. Ali, “Corpus-based method in language learning and teaching,” *International Journal of Research and Analytical Reviews*, vol. 6, no. 2, pp. 473–476, 2019.

## Article

# Cross-Corpus Speech Emotion Recognition Based on Multi-Task Learning and Subdomain Adaptation

Hongliang Fu <sup>1,2,3,\*</sup>, Zhihao Zhuang <sup>1,2</sup>, Yang Wang <sup>1,2</sup>, Chen Huang <sup>1,2</sup> and Wenzhuo Duan <sup>1,2</sup>

<sup>1</sup> College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China

<sup>2</sup> Henan Engineering Laboratory of Grain IOT Technology, Henan University of Technology, Zhengzhou 450001, China

<sup>3</sup> Key Laboratory of Food Information Processing and Control, Ministry of Education, Henan University of Technology, Zhengzhou 450001, China

\* Correspondence: jackfu\_zz@163.com

**Abstract:** To solve the problem of feature distribution discrepancy in cross-corpus speech emotion recognition tasks, this paper proposed an emotion recognition model based on multi-task learning and subdomain adaptation, which alleviates the impact on emotion recognition. Existing methods have shortcomings in speech feature representation and cross-corpus feature distribution alignment. The proposed model uses a deep denoising auto-encoder as a shared feature extraction network for multi-task learning, and the fully connected layer and softmax layer are added before each recognition task as task-specific layers. Subsequently, the subdomain adaptation algorithm of emotion and gender features is added to the shared network to obtain the shared emotion features and gender features of the source domain and target domain, respectively. Multi-task learning effectively enhances the representation ability of features, a subdomain adaptive algorithm promotes the migrating ability of features and effectively alleviates the impact of feature distribution differences in emotional features. The average results of six cross-corpus speech emotion recognition experiments show that, compared with other models, the weighted average recall rate is increased by 1.89%~10.07%, the experimental results verify the validity of the proposed model.

**Citation:** Fu, H.; Zhuang, Z.; Wang, Y.; Huang, C.; Duan, W. Cross-Corpus Speech Emotion Recognition Based on Multi-Task Learning and Subdomain Adaptation. *Entropy* **2023**, *25*, 124. <https://doi.org/10.3390/e25010124>

Academic Editors: Andrea Prati, Luis Javier García Villalba and Vincent A. Cicirello

Received: 26 December 2022

Revised: 3 January 2023

Accepted: 4 January 2023

Published: 7 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Speech is a very valuable research object to realize intelligent interaction today. Through speech communication, human beings can not only obtain the speaker's semantic information, but also perceive the speaker's emotional state, gender, age and other paralinguistic content [1]. In the middle of the 20th century, human–computer interaction (HCI) systems mainly conveyed instructions to computers through the mouse and keyboard, and did not have the ability to perceive speech emotional information. In order to improve the intelligence of a computer and meet the comfortable and convenient needs of users, it is particularly important to make the computer have the speech-emotional information perception ability like human beings. In this context, researchers began to explore the emotional information processing of speech.

Speech Emotion Recognition (SER) first began using acoustic statistical features to classify emotions [2] in the 1980s, these acoustic features are still widely used in speech analysis [3,4]. With the rapid development of artificial intelligence in the 21st century, speech emotion recognition technology has been widely used in various fields, including call quality detection in a customer service center, speech assistants and auxiliary diagnoses. Therefore, SER has very important practical application research value.

In real application scenarios, different corpora have different recording environments, personnel gender, age distribution and languages, resulting in great variations in feature distribution among different corpora, which makes it difficult for models trained based on a single corpus to achieve good recognition results on new speech signal [5]. Speech emotion recognition also has some limitations in other aspects. For example, in the case of strong background noise, emotional information is difficult to be effectively recognized. Therefore, many scholars try to supplement it with other aspects, including facial emotion recognition [6–8] and physiological signal emotion recognition [9,10].

In order to further enhance the generalization of the speech emotion recognition model, the main contributions of this work are summarized as follows:

1. The proposed method uses multi-task learning to help the network extract speech features, which is more robust than the features obtained only using emotional recognition tasks.
2. A subdomain transfer learning method is proposed, which can reduce the negative transfer in the whole local adaptation process more than the global adaptation method.
3. In the ablation experiment and the evaluation compared with other algorithms, the proposed method has achieved performance leadership in most cross-corpus schemes.

## 2. Related Work

At present, the recognition rate of speech emotion recognition has reached the level of human recognition, but this can only be achieved under the condition of acoustic laboratory and some specific emotion corpus. When the training data and test data come from different corpora, the model performance often suffers a serious decline. Many researchers propose cross-corpus algorithms to solve the data discrepancy to improve the model performance.

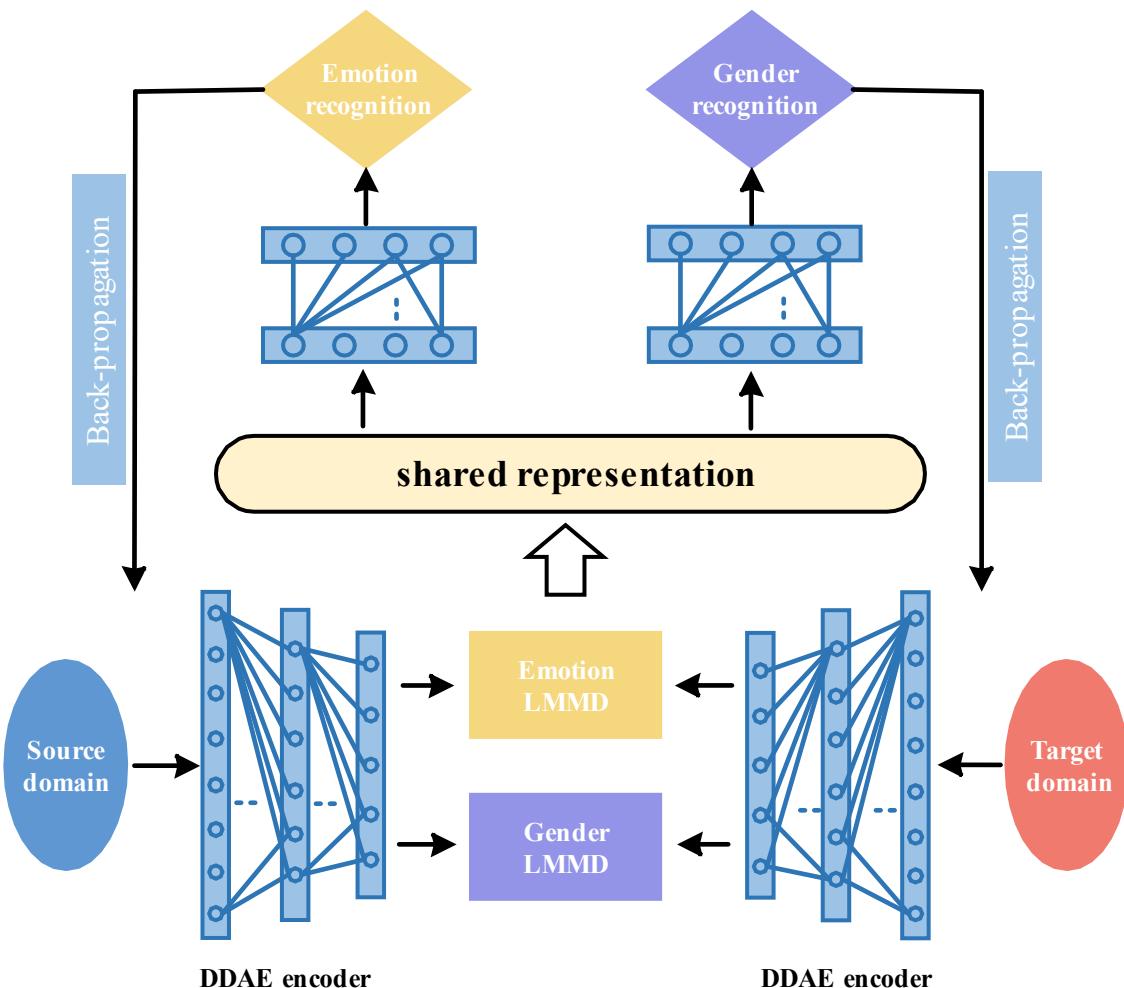
Deng et al. [11] used unsupervised learning methods of denoising auto-encoder and domain adaptive technology to solve the inherent difference between the training set and the test set. Huang et al. [12] proposed a new feature transfer method based on PCANet to learn the emotional features of unlabeled data by measuring the distribution offset between training data and test data. Zong et al. [13] proposed a domain adaptive least squares regression model. The least squares regression model was trained by adding regularization constraints of source domain data and a group of target domain data to the objective function to improve cross-corpus recognition performance. In addition, the subspace learning algorithm has also achieved satisfactory results in the cross-corpus SER. For example, Liu et al. [14] proposed a domain adaptive subspace learning method to learn the projection matrix and convert the speech signal from the original feature space to the label subspace; Song et al. [15] proposed a transfer linear subspace learning framework, and used the nearest neighbor graph algorithm to measure the similarity between different corpora, so as to achieve cross-corpus speech emotion recognition research; Luo et al. [16] extracted the source domain data and target domain data to obtain the shared subspace feature representation and two independent feature representations, and used the orthogonal constraint method to eliminate the redundancy of shared features and independent features, while minimizing the difference between the conditional distribution and marginal distribution of the source domain and target domain in the shared subspace. Finally, they achieved high recognition rates in 30 sets of cross-corpus emotion recognition experiments. In addition, the combination of deep learning and domain adaptation to solve cross-corpus speech emotion recognition problems has gradually become a new research focus. For example, Liu et al. [17] used the depth convolution neural network and the maximum mean discrepancy (MMD) to perform feature migration and achieve cross-corpus speech emotion recognition.

Therefore, the influencing factors of cross-corpus SER system performance can be summarized as follows:

1. To obtain the emotional information with strong representation ability in speech feature. Human speech contains a variety of paralinguistic information in addition to semantic information, such as mood, gender, emotion, but the ideal speech emotional feature should be independent of the speaker, semantics, language and other objective factors, and reflect emotional information as effectively as possible, which puts forward higher requirements for the generalization of emotional features of the cross-corpus SER system.
2. To effectively measure the distribution discrepancy of features. In cross-corpus SER research, researchers mostly use the emotion feature measurement criteria based on the global feature area [12,13,17], and only measure the distance between two emotion vector matrices representing the source domain and target domain, ignoring the differences of different emotion features in the field, which may lead to the confused transfer of similar emotion information, such as happy and surprise, anger and disgust, which is not conducive to the subsequent emotion classification.

### 3. Model Framework

Multi-task learning can improve the generalization of the main task recognition performance. This chapter introduces a cross-corpus SER model based on Multi-task learning and subdomain adaptation (MTLSA), as shown in Figure 1. First, it is confirmed that the main recognition task of MTLSA is emotion recognition, while the auxiliary recognition task is gender recognition. Secondly, in the aspect of feature processing, the model MTLSA in this chapter uses the deep denoising auto-encoder (DDAE) network as the task-sharing network. On this basis, task-specific layers with attribute dependency are added, so that when the network learns the shared features, it allows each task-specific layer to optimize its own attribute parameters to improve performance. Then, in the low dimensional emotional features output by the DDAE code, the whole region is divided into emotional subdomain space and gender subdomain space according to emotional labels and gender labels, and the subdomain adaptation algorithm based on the local maximum mean discrepancy (LMMD) [18] is used to reduce the feature distribution distance between the source domain and target domain. Finally, the cross entropy loss calculation is performed using the emotion label and gender label information of the source domain, and the MTLSA is constrained by the feature reconstruction loss and feature distribution distance measurement loss. The MTLSA multi-task learning module and subdomain adaptation will be described in detail in Sections 3.1 and 3.2, and the MTLSA training and recognition process will be described in Section 3.3.



**Figure 1.** Overall Framework of Multi-task Learning and Subdomain Adaptive Model.

### 3.1. Multi-Task Learning

In the cross-corpus SER research, in order to further reduce the discrepancy in the distribution of emotional features and improve the generalization of the system, the multi-task learning mechanism is introduced to eliminate the emotional differences caused by gender factors, so as to learn more common emotional information between different fields. In this section, MTLSA performs feature matching under the multi-task learning mechanism based on hyper-parameter sharing. The sharing network of the emotion recognition task and gender recognition task is DDAE. It has been verified that the reconstructed features can effectively compress feature dimensions and remove feature redundancy. On this basis, the model adds noise to the DAE and builds a DDAE network to extract common emotional features from the source domain and target domain to enhance system robustness.

The sample features of the source domain are given as follows:  $X_S = [x_1^S, \dots, x_{n_s}^S] \in R^{d \times n_s}$ , the emotional category label of the source domain sample is  $Y_S = [y_1, \dots, y_{n_s}] \in R^{C \times n_s}$ , the gender category label of the source domain sample is  $Y_G = [y_1, \dots, y_{n_s}] \in R^{2 \times n_s}$ , and the sample features of the target domain is  $X_T = [x_1^T, \dots, x_{n_t}^T] \in R^{d \times n_t}$ . Among this,  $n_s$  and  $n_t$  represent the number of samples in the source domain and target domain, respectively,  $d$  represent the emotional feature

dimension of each speech sample, and  $C$  represent the number of emotional categories. DDAE is used for redundant compression of speech features to obtain common emotional features with robustness and effective representation. First, add the noise with the normal distribution (mean value is 0, variance is 1) in the source domain  $X_s$  and target domain  $X_t$ . Then, low-level features with noise are input into DDAE, and the source domain and target domain feature vectors decoded by DDAE are represented as  $\tilde{X}_s$  and  $\tilde{X}_t$ , respectively. Therefore, the loss function of the DDAE network processing features includes the reconstruction loss function  $L_s$  of  $X_s$  and the reconstruction loss function  $L_t$  of  $X_t$ , which are, respectively, expressed as:

$$L_s = (X_s, \tilde{X}_s) = \sum_{i=1}^{n_s} \|x_i^s - \tilde{x}_i^s\|^2 \quad (1)$$

$$L_t = (X_t, \tilde{X}_t) = \sum_{i=1}^{n_t} \|x_i^t - \tilde{x}_i^t\|^2 \quad (2)$$

The task-specific layer consists of two independent full connection layers, which input the results into the softmax layer and output the emotion labels. In the cross-corpus research based on domain adaptation, the main task emotion recognition and the auxiliary task gender recognition will use the source domain real label information and the source domain softmax prediction label to calculate the cross entropy as a loss function to constrain the parameter update of different tasks at the specific layer. The prediction probabilities of the emotion category and gender category of the source domain samples are expressed as  $p_i^s = [p_1^s, \dots, p_{n_s}^s]$  and  $p_i^g = [p_1^g, \dots, p_{n_g}^g]$ , respectively, and the cross entropy is calculated with the ground truth, respectively, and the emotion classification loss function  $L_y$  and gender classification loss function  $L_g$  of the source domain are obtained.

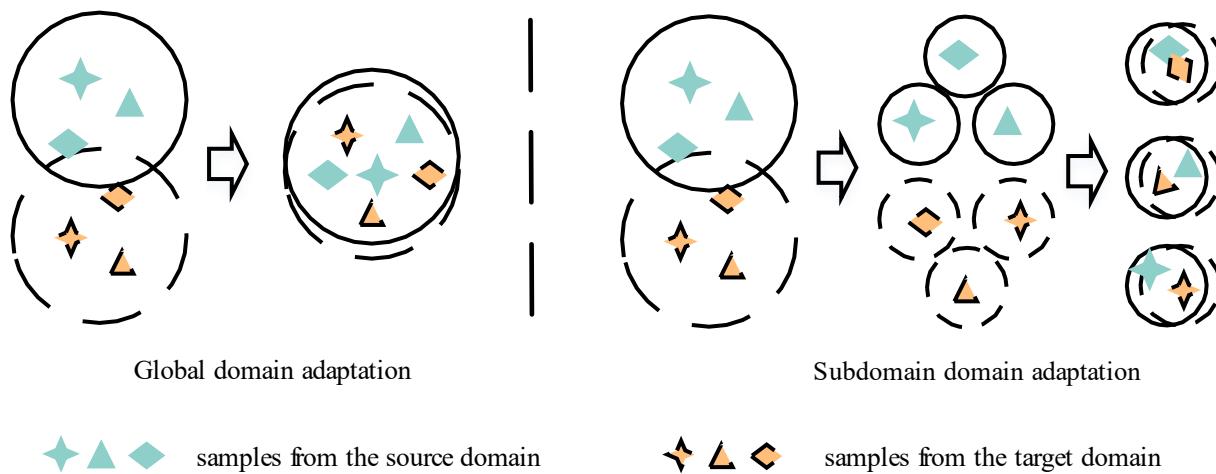
$$L_y(Y_s, p_i^s) = -\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{c=1}^C y_i^s \log(p_i^s) \quad (3)$$

$$L_g(Y_g, p_i^g) = \frac{1}{n_s} \sum_{i=1}^{n_s} -[y_i^g \cdot \log(p_i^g) + (1 - y_i^g) \cdot \log(1 - p_i^g)] \quad (4)$$

### 3.2. Subdomain Adaptation

To learn common emotional information through gender recognition tasks by multi-task learning. At the same time, it uses a subdomain adaptive algorithm based on Local Maximum Mean Discrepancy (LMMD) to measure the feature distributions discrepancy between the source domain and the target domain, as shown in Figure 2, so as to reduce the emotional differences and gender differences in speech and improve the generalization of the system. The MTLSA model divides the low dimensional features output by the DDAE encoder into independent emotion subdomain space and gender subdomain space according to the emotion labels and gender labels of the source domain, and the emotion prediction label and gender prediction label of the target domain, so as to achieve accurate emotion feature alignment and gender feature alignment.

In the emotion subdomain space, the emotion features output by the source domain and target domain through the DDAE encoder are represented as  $X'_s = [x_1'^s, \dots, x_{n_s}'^s] \in R^{d' \times n_s}$  and  $X'_t = [x_1'^t, \dots, x_{n_t}'^t] \in R^{d' \times n_t}$ , respectively, and the feature distribution is aligned through LMMD, and the measured distribution distance can be used as loss function  $L_{DE}$  to continuously reduce during the training process.



**Figure 2.** Differences between subdomain adaptation and global domain adaptation.

$$L_{DE} = \frac{1}{C} \sum_{c=1}^C \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \mu_{i,c}^S \delta(x_i'^S) - \frac{1}{n_T} \sum_{i=1}^{n_T} \mu_{i,c}^T \delta(x_i'^T) \right\|_H^2 \quad (5)$$

Among them,  $H$  is the reproducing kernel hilbert space (RKHS), and  $\delta(\cdot)$  represents the kernel function that maps emotional features to RKHS.  $\mu_{i,c}^S$  and  $\mu_{i,c}^T$ , respectively, represent the weight vectors of  $x_i'^S$  and  $x_i'^T$  belonging to the emotion category. The weight  $\mu_{i,c}$  of sample feature  $x_i'$  is calculated as  $\mu_{i,c} = y_{i,c} / \sum_{(x_j, y_j) \in D} y_{j,c}$ . It is worth noting that the emotional label  $y_{i,c}^S$  of the sample features in the source domain is known, while the target domain cannot directly obtain  $y_{i,c}^T$ . Here, softmax outputs the sample feature probability of the target domain to generate the pseudo tag  $y_{i,c}^T$ .

In the gender subdomain space, the gender features of the source domain and target domain encoded by DDAE are  $X'_{SG} = [x_1'^{SG}, x_2'^{SG}, \dots, x_{n_S}'^{SG}] \in R^{d' \times n_S}$  and  $X'_{TG} = [x_1'^{TG}, x_2'^{TG}, \dots, x_{n_T}'^{TG}] \in R^{d' \times n_T}$ , respectively. Similarly, gender features are aligned by LMMD, and the metric distance is expressed as  $L_{DG}$ .

$$L_{DG} = \frac{1}{M} \sum_{m=1}^M \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \beta_{i,M}^S \delta(x_i'^{SG}) - \frac{1}{n_T} \sum_{i=1}^{n_T} \beta_{i,M}^T \delta(x_i'^{TG}) \right\|_u^2 \quad (6)$$

Wherein,  $\beta_{i,M}^S$  and  $\beta_{i,M}^T$ , respectively, represent the weight vectors of source domain feature  $x_i'^{SG}$  and target domain  $x_i'^{TG}$  that belong to the gender category  $M$ .  $M = 2$ , like formula (5),  $y_{i,M}^T$  cannot be directly obtained. The target domain samples need to generate pseudo label information  $y_{i,M}^T$  through softmax output.

### *3.3. Model Training and Identification*

The total loss function of the MTLSA can be expressed as:

$$L_{SUM} = a \cdot L_S + b \cdot L_T + c \cdot L_Y + d \cdot L_G + e \cdot L_{DE} + f \cdot L_{DG} \quad (7)$$

Among them,  $\{L_s, L_T, L_Y, L_G, L_{DE}, L_{DG}\}$  represents the reconstruction loss of source domain sample features, the reconstruction loss of target domain sample features, the emotional classification loss function of source domain sample features, the gender classification loss function of source domain sample features, the emotional feature

distribution distance, and the gender feature distribution distance, respectively.  $\{a,b,c,d,e,f\}$  represents the loss weight coefficient of  $\{L_S, L_T, L_Y, L_G, L_{DE}, L_{DG}\}$ , respectively, and the values of  $a+b+c+d+e+f=1$  and  $\{a,b,c,d,e,f\}$  are determined through debugging.

In the recognition stage, the target domain samples are used as the test corpus, and the emotion features are extracted from the trained network. After the softmax layer outputs the prediction probability, the label information corresponding to the maximum probability value is selected as the sample recognition result, and the emotion labels of the target domain samples are finally output.

## 4. Experimental Setup

### 4.1. Corpus

In order to ensure the consistency of the experiment and the fairness of the evaluation of the experimental indicators, the proposed method uses the most widely used corpus for evaluation. Three public corpora, Berlin [19], eINTERFACE [20], and CASIA [21] are selected as the corpora of the experiment. Berlin is recorded by five male and five female actors simulating anger, boredom, disgust, fear, neutral and sad. eINTERFACE included 34 male and eight female subjects anger, disgust, fear, happy, sad and surprise. CASIA contains the anger, fear, happy, neutral, sad and surprise of two male and three female speakers. In order to carry out cross-corpus research, we selected the samples of source domain and target domain that come from different corpora, but the emotional labels of the two corpora are the same. Therefore, three samples of three corpora need to be reselected to meet the experimental requirements.

In terms of emotion recognition, the same emotions of Berlin and eINTERFACE are disgust, anger, sad, fear and happy, and the sample numbers are 375 and 1072, respectively. The same emotions of eINTERFACE and CASIA are surprise, anger, sad, fear and happy, and the sample numbers are 1072 and 1000, respectively. The same emotions of Berlin and CASIA are neutral, anger, sad, fear and happy, with 408 and 1000 samples selected, respectively.

In identifying gender, we need to make gender tags of three corpora. The samples of the material corpus used in the two identification tasks are exactly the same, only the label types are different. Among them, the number of male samples in Berlin and eINTERFACE is 159 and 885, respectively, and the number of female samples is 216 and 187, respectively; eINTERFACE and CASIA. The number of male samples in the library is 847 and 500, respectively, and the number of female samples is 225 and 500, respectively. The number of male samples in Berlin and CASIA is 187 and 500, respectively, and the number of female samples is 221 and 500, respectively. Table 1 summarizes the corpus information used for cross-corpus identification.

**Table 1.** Corpora information for cross-corpus identification.

Corpus	Emotion Recognition Task		Gender Identification Task	
	Num of Samples	Emotional Tags	Male Samples	Female Samples
Berlin	375	Anger, Sad, Fear,	159	216
eINTERFACE	1072	Happy, Disgust	885	187
CASIA	1000	Anger, Sad, Fear,	500	500
eINTERFACE	1072	Happy, Surprise	847	225
Berlin	408	Anger, Sad, Fear,	187	221
CASIA	1000	Happy, Neutral	500	500

#### 4.2. Extract Speech Features

This section uses the emotional feature set specified in the INTERSPEECH2010 emotional challenge as the speech of all emotional feature set. Based on 34 LLDs, 1428 dimensional features are obtained by using 21 statistical functions. Secondly, on the basis of LLDs and delta coefficients of four treble, 152 dimensional features are obtained by using 19 statistical functions. Then, add the start time and duration of the speech into it. Finally, a total of 1582 dimensional artificial statistical emotional feature set is obtained [22]. Use the openSMILE tool [23] to extract 1582 dimension features of three corpora in Table 1. In addition, these speech features need to be normalized before input network training to compress the eigenvalues in the (0,1) range.

#### 4.3. Experimental Scheme

Choose between two corpora randomly from the three corpora, and choose speech samples with the same emotion between the two corpora to design the experimental scheme, one of which is used as the source domain corpus, the other as the target domain corpus. Using the letters B, E and C to represent Berlin, eINTERFACE and CASIA, respectively, six cross-corpus speech emotion recognition experimental schemes are designed, which are E→B, B→E, E→C, C→E, B→C, C→B. Table 2 summarizes the source domain and target domain of different cross-corpus experimental schemes, as well as the cross-corpus identification tasks of each scheme.

In the six experimental schemes, the learning rate and batch size of MTLSA are set to 0.000001 and 100, respectively, the network optimizer and classifier use Adam and softmax, respectively, and the model is iteratively trained 300 times. In the training process, the weight coefficients {a, b, c, d, e, f} of the six loss functions of the model are [0.05, 0.05, 0.6, 0.1, 0.1, 0.1]. For DDAE, the sizes of hidden layer neuron nodes are 1200, 900, 256, 900 and 1200, respectively, where the encoding and decoding stages use the ELU function and Sigmoid function, respectively. In addition, each layer of DDAE adds a Batch Normal (BN) layer and a Dropout layer. For task-specific layers in multi-task learning, the hidden layer neuron node size is 256.

**Table 2.** Six cross-corpus experimental schemes and identification tasks.

Scheme	Source Domain	Target Domain	Cross-Corpus Identification
E→B	eINTERFACE	Berlin	Anger, Sad, Fear, Happy, Dis-
B→E	Berlin	eINTERFACE	gust
E→C	eINTERFACE	CASIA	Anger, Sad, Fear, Happy, Sur-
C→E	CASIA	eINTERFACE	prise
B→C	Berlin	CASIA	Anger, Sad, Fear, Happy, Neu-
C→B	CASIA	Berlin	tral

### 5. Analysis of Experimental Results

#### 5.1. Analysis of Ablation Experiment

This section conducts ablation experiments to evaluate the effectiveness of different modules in MTLSA, and sets up two ablation models. (1) MTLSA\_L indicates that the proposed model MTLSA only uses the LMMD algorithm for emotional feature distribution alignment and gender feature distribution alignment, and does not use multi-task learning; (2) MTLSA\_M means that MTLSA only uses the multi-task learning framework to learn shared features, and does not use the LMMD algorithm for feature alignment. In the six cross-corpus experimental schemes, the experimental results of two ablation models and MTLSA are shown in Table 3.

**Table 3.** WAR of different ablation models in six cross-corpus schemes (%).

Model	E→B	B→E	E→C	C→E	B→C	C→B
MTLSA_L	36.80	24.44	32.90	23.23	30.10	39.95
MTLSA_M	55.73	30.60	34.40	30.32	39.30	53.94
MTLSA	<b>57.60</b>	<b>34.12</b>	<b>35.21</b>	<b>31.52</b>	<b>41.90</b>	<b>56.86</b>

From Table 3, it can be seen that the WAR of the proposed model MTL SA in this chapter is higher than those of other ablation models under the six schemes, indicating that it is an effective practice for MTL SA to combine multi-task learning with subdomain adaptive feature transfer. From the WAR of MTL SA\_L and MTL SA\_M, it can be seen that MTL SA only uses a deep denoising auto-encoder to extract common features, and on this basis, LMMD is used to measure the distribution distance of emotional features and gender feature distribution distance, and the system performance of using LMMD to measure the distribution distance of emotional features is poor, while the multi-task learning architecture is used to extract common features, and the use of auxiliary tasks to learn emotion-related information is beneficial to obtain more emotional features, effectively reducing the feature distribution distance between the source domain and the target domain. Multi-task learning and subdomain adaptation are both forms of transfer learning, and the fusion of the two can extract salient emotional features and effectively improve the generalization of the system.

### 5.2. Comparative Experimental Analysis

In this section, some state-of-the art cross-corpus SER models are used for comparison to evaluate the performance of MTL SA, including Transfer Sparse Discriminant Subspace Learning (TSDSL) [22], Deep Belief Network and Back Propagation (DBN+BP) [24], Domain Adaptive Subspace Learning (DoSL) [14]. At the same time, PCA+SVM is selected as the reference algorithm for the experiment, and the SVM classifier adopts a linear kernel function. Table 4 shows the WAR results of the MTL SA and other advanced models and benchmark models in six cross-corpus recognition schemes.

It can be seen from Table 4 that the WAR of the proposed model MTL SA is higher than PCA+SVM, TSDSL and DBN+BP in six cross-corpus schemes, indicating that multi-task learning combined with subdomain adaptive reduction in feature distribution differences is advanced. Among them, TSDSL only reduces the feature distribution distance in the global domain emotion space, and ignores the connection between more fine-grained emotion categories, and the model in this chapter uses emotion labels and gender labels to divide the feature space into independent subdomain space, considering the confusing alignment influence of different emotion information, and accurately aligning the feature distribution of the same emotion and gender. DBN+BP belongs to the application of deep learning with the proposed model, but DBN+BP only uses the basic feature processing method, and does not use the correlation feature transfer learning algorithm to train the cross-corpus emotion classifier, so the cross-corpus recognition effect is not ideal, DoSL uses subspace learning methods, but only features reduction and dimension selection, and does not achieve accurate domain alignment. It is difficult to effectively improve the generalization of the cross-corpus speech emotion recognition model.

**Table 4.** WAR of comparison model in six cross-corpus schemes (%).

Model	E→B	B→E	E→C	C→E	B→C	C→B	Average
PCA+SVM	50.85	33.68	28.60	27.80	33.60	43.87	36.40
TSDSL [22]	50.67	<b>35.47</b>	32.50	33.28	37.40	56.60	40.98
DBN+BP [24]	26.67	32.28	24.20	31.04	35.80	46.81	32.80
DoSL [14]	49.58	30.64	35.20	<b>33.90</b>	35.77	<b>57.51</b>	40.43
MTLSA	<b>57.60</b>	34.12	<b>35.21</b>	31.52	<b>41.90</b>	56.86	<b>42.87</b>

Compared with the above single task-learning method, the structure of multi-task learning is generally composed of shared modules and task modules. The shared modules contain shared network parameters, and the task modules contain different tasks that the network needs to complete. Multi-task learning trains multiple tasks in parallel by sharing network layer parameters, and finally enables a single network to achieve multiple functions, which is also the key to improving model generalization. It can be concluded that gender is an important factor affecting the performance of cross-corpus speech emotion recognition, and learning common gender information while extracting common emotion information can effectively alleviate the gender difference in emotional features and help further reduce the feature distribution distance between the source domain and the target domain.

## 6. Conclusions

This paper proposed a cross-corpus speech emotion recognition model based on multi-task learning and subdomain adaptation to alleviate the impact of gender factors on emotion recognition. The model takes emotion recognition as the main task, gender recognition as the auxiliary task, and uses the deep denoising auto-encoder as the shared network of the multi-task learning framework to extract the emotional common information and gender common information with strong representation ability. LMMD-based subdomain adaptive algorithm is used to constrain learning emotion and gender features, and further, obtain shared information. From a large number of experimental results, the model proposed in this chapter can not only effectively reduce the difference in feature distribution between the source domain and the target domain, but also alleviate the impact of gender attributes on emotion recognition, providing a new idea for solving the problem of cross-corpus speech emotion recognition.

**Author Contributions:** Conceptualization, H.F.; Data curation, Y.W., C.H. and W.D.; Formal analysis, Z.Z.; Funding acquisition, H.F.; Investigation, H.F., Z.Z. and Y.W.; Software, Z.Z.; Supervision, H.F.; Validation, Z.Z.; Writing—original draft, H.F. and Z.Z.; Writing—review and editing, Y.W., C.H. and W.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research project was founded in part by National Natural Science Foundation of China (Grant No. 61975053), Natural Science Project of Henan Education Department (Grant No. 22A510013, Grant No. 22A520004 and Grant No. 22A510001), Start-up Fund for High-level Talents of Henan University of Technology (No. 2018BS037).

**Institutional Review Board Statement:** Not applicable

**Data Availability Statement:** Not applicable

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Alisamir, S.; Ringeval, F. On the Evolution of Speech Representations for Affective Computing: A brief history and critical overview. *IEEE Signal Process. Mag.* **2021**, *38*, 12–21.
2. Malik, M.; Malik, M.K.; Mehmood, K.; Makhdoom, I. Automatic speech recognition: A survey. *Multimed. Tools Appl.* **2021**, *80*, 9411–9457.
3. Sitaula, C.; He, J.; Priyadarshi, A.; Tracy, M.; Kavehei, O.; Hinder, M.; Hinder, M.; Withana, A.; McEwan, A.; Marzbanrad, F. Neonatal Bowel Sound Detection Using Convolutional Neural Network and Laplace Hidden Semi-Markov Model. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2022**, *30*, 1853–1864. <https://doi.org/10.1109/TASLP.2022.3178225>.
4. Burne, L. et al. Ensemble Approach on Deep and Handcrafted Features for Neonatal Bowel Sound Detection. *IEEE J. Biomed. Health Inform.* **2022**. <https://doi.org/10.1109/JBHI.2022.3217559>.
5. Lee, S. Domain Generalization with Triplet Network for Cross-Corpus Speech Emotion Recognition. In Proceedings of the IEEE Spoken Language Technology Workshop, Shenzhen, China, 19–22 January 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 389–396.
6. Antoniadis, P.; Filntisis, P.P.; Maragos, P. Exploiting Emotional Dependencies with Graph Convolutional Networks for Facial Expression Recognition. In Proceedings of the 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), Jodhpur, India, 15–18 December 2021; pp. 1–8. <https://doi.org/10.1109/FG52635.2021.9667014>.

7. Ryumina, E.; Dresvyanskiy, D.; Karpov, A. In search of a robust facial expressions recognition model: A large-scale visual cross-corpus study. *Neurocomputing* **2022**, *514*, 435–450.
8. Savchenko, A.V.; Savchenko, L.V.; Makarov, I. Classifying Emotions and Engagement in Online Learning Based on a Single Facial Expression Recognition Neural Network. *IEEE Trans. Affect. Comput.* **2022**, *13*, 2132–2143. <https://doi.org/10.1109/TAFFC.2022.3188390>.
9. Du, G.; Su, J.; Zhang, L.; Su, K.; Wang, X.; Teng, S.; Liu, P.X. A Multi-Dimensional Graph Convolution Network for EEG Emotion Recognition. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3204314.
10. Liu, S.; Wang, X.; Zhao, L.; Li, B.; Hu, W.; Yu, J.; Zhang, Y. 3DCANN: A spatio-temporal convolution attention neural network for EEG emotion recognition. *IEEE J. Biomed. Health Inform.* **2021**, *26*, 5321–5331.
11. Deng, J.; Zhang, Z.; Eyben, F.; Schuller, B. Autoencoder-based unsupervised domain adaptation for speech emotion recognition. *IEEE Signal Process. Lett.* **2014**, *21*, 1068–1072.
12. Huang, Z.; Xue, W.; Mao, Q.; Zhan, Y. Unsupervised domain adaptation for speech emotion recognition using PCANet. *Multimed. Tools Appl.* **2017**, *76*, 6785–6799.
13. Zong, Y.; Zheng, W.; Zhang, T.; Huang, X. Cross-corpus speech emotion recognition based on domain-adaptive least-squares regression. *IEEE Signal Process. Lett.* **2016**, *23*, 585–589.
14. Liu, N.; Zong, Y.; Zhang, B.; Liu, L.; Chen, J.; Zhao, G.; Zhu, J. Unsupervised cross-corpus speech emotion recognition using domain-adaptive subspace learning. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 5144–5148.
15. Song, P. Transfer linear subspace learning for cross-corpus speech emotion recognition. *IEEE Trans. Affect. Comput.* **2019**, *10*, 265–275.
16. Luo, H.; Han, J. Nonnegative matrix factorization based transfer subspace learning for cross-corpus speech emotion recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 2047–2060.
17. Liu, J.; Zheng, W.; Zong, Y.; Lu, C.; Tang, C. Cross-corpus speech emotion recognition based on deep domain-adaptive convolutional neural network. *IEICE Trans. Inf. Syst.* **2020**, *103*, 459–463.
18. Zhu, Y.; Zhuang, F.; Wang, J.; Ke, G.; Chen, J.; Bian, J.; Xiong, H.; He, Q. Deep subdomain adaptation network for image classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1713–1722.
19. Burkhardt, F.; Paeschke, A.; Rolfs, M.; Sendlmeier, W.F.; Weiss, B. A-corpus of German emotional speech. In Proceedings of the Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, 4–8 September 2005; Volume 5, pp. 1517–1520.
20. Martin, O.; Kotsia, I.; Macq, B.; Pitas, I. The eINTERFACE'05 audio-visual emotion-corpus. In Proceedings of the 22nd International Conference on Data Engineering Workshops, Atlanta, GA, USA, 3–7 April 2006; IEEE: Piscataway, NJ, USA, 2006; p. 8.
21. Tao, J.; Liu, F.; Zhang, M.; Jia, H. Design of speech corpus for mandarin text to speech. In Proceedings of the Blizzard Challenge 2008 Workshop, Brisbane Australia, 20 September 2008.
22. Zhang, W.; Song, P. Transfer sparse discriminant subspace learning for cross-corpus speech emotion recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *28*, 307–318.
23. Eyben, F.; Wöllmer, M.; Schuller, B. Opensmile: The munich versatile and fast open-source audio feature extractor. In Proceedings of the 18th ACM International Conference on Multimedia, Firenze Italy, 25–29 October 2010; pp. 1459–1462.
24. Latif, S.; Rana, R.; Younis, S.; Qadir, J.; Epps, J. Transfer learning for improving speech emotion classification accuracy. *arXiv*, **2018** preprint arXiv:1801.06353.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

METHODOLOGY

Open Access



# Cross-corpus speech emotion recognition using subspace learning and domain adaption

Xuan Cao<sup>1</sup>, Maoshen Jia<sup>1\*</sup> , Jiawei Ru<sup>1</sup> and Tun-wen Pai<sup>2</sup>

## Abstract

Speech emotion recognition (SER) is a hot topic in speech signal processing. When the training data and the test data come from different corpus, their feature distributions are different, which leads to the degradation of the recognition performance. Therefore, in order to solve this problem, a cross-corpus speech emotion recognition method is proposed based on subspace learning and domain adaptation in this paper. Specifically, training set data and the test set data are used to form the source domain and target domain, respectively. Then, the Hessian matrix is introduced to obtain the subspace for the extracted features in both source and target domains. In addition, an information entropy-based domain adaption method is introduced to construct the common space. In the common space, the difference between the feature distributions in the source domain and target domain is reduced as much as possible. To evaluate the performance of the proposed method, extensive experiments are conducted on cross-corpus speech emotion recognition. Experimental results show that the proposed method achieves better performance compared with some existing subspace learning and domain adaptation methods.

**Keywords:** Speech emotion recognition, Cross-corpus, Subspace learning, Domain adaption

## 1 Introduction

There are many ways for people to express emotions, such as through speech, actions, and facial expressions. Speech is an important way to express emotions among these ways, because it contains riches emotions, such as happy, angry, and sad. Speakers can deliver their intentions through different tones, volumes, or content. How to judge a speaker's emotion through speech becomes crucial. Therefore, speech emotion recognition (SER) is an important branch of many modal affective computing, and it is also an important part of speech recognition. With the development of SER, it has been applied in the fields of psychotherapy, human-computer interaction, etc. According to the results of SER, the machine can generate appropriate responses for the user in an interactive environment. Therefore, SER is one of the

most important technologies for human-computer interaction [1–4].

The semantic-based methods are an important class of SER methods, because emotions can be expressed effectively by semantics. If the speakers use emotive words to communicate with others, then we can directly judge the emotion from the semantics of the words. Therefore, semantic-based research gradually began to develop. A multi-classifier emotion recognition model based on prosodic information and semantic labels is introduced in [5]. Similarly, the semantic labels and the non-verbal audio in speech, such as onomatopoeia such as crying, laughter, or sighing, are used in SER [6]. Subsequently, temporal and semantic coherence is introduced for SER [7]. In addition, the model of bimodal SER from acoustic and linguistic information fusion is proposed [8].

Although semantics understanding is simply for humans, it is a complex process for machines. Therefore, more research is currently aimed at speech

\*Correspondence: [jiamashen@bjut.edu.cn](mailto:jiamashen@bjut.edu.cn)

<sup>1</sup> Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

Full list of author information is available at the end of the article



© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

features that are easily understood by machines, which is also important for SER. Compared with semantic information, speech features are more abstract. But they are very important for expressing the speaker's emotions. The main features used in SER are divided into acoustic features and spectral features. The acoustic features include intensity, pitch, and timbre. Features like energy, Mel-frequency cepstral coefficients (MFCC), linear prediction coefficients (LPC), and fundamental frequency are called spectral features. The features such as pitch, MFCC, formant, intensity, and chroma are adopted for SER [9, 10]. Also, the pitch, spectrum, and formant are combined with semantic information for recognizing emotions in [5]. To improve the robustness of SER, some methods have been used to process the features. Specifically, PCA is adopted to reduce the dimensionality of the features [11], and a statistical method is utilized to find robust spectral features [12].

In practical scenarios, the speaker's emotion is very complex. The speaker may have multiple emotions at the same time, rather than a single emotion, or the emotion expressed by the speaker is inconsistent with the actual emotion. It makes SER difficult. There is also research proposed for complex emotions. A circular continuous dimensional model to describe an emotion, called valence-arousal model (VA) was proposed in [13, 14]. The model no longer regards emotions as discrete but uses two-dimensional coordinates to describe the continuous distribution of emotions. The PAD emotional model was shown in [15, 16], which has P (pleasure), A (arousal), and D (dominance) values to represent all emotional states. In addition, based on the emotional probability distribution, an ambiguous label is proposed to solve the inconsistency problem in ambiguous emotional cognition [17].

Another problem in SER is how to recognize emotions. To this end, some machine learning methods were adopted to recognize emotions, such as support vector machine (SVM) [18], hidden Markov model (HMM) [19], and Gaussian mixed model (GMM) [20]. In recent years, with the rapid development of deep learning, various neural network structures have been introduced in SER. From convolutional neural networks (CNN) [21], recurrent neural networks (RNN) [22], back propagation neural network (BPNN) [23], and deep neural network (DNN) [24] to sequential capsule networks [25] and adversarial data augmentation network [26], they are both used for SER. A segment-based iterative self-learning enhanced speech emotion recognition model is proposed in [27]. The above algorithms perform well in traditional SER, and

the recognition accuracy of some algorithms can even reach more than 80% in some corpora settings. In the actual scene, the speech signals do not belong to a specific corpus, which are recorded in different scenes. The speech data is also affected by language, gender, speaking styles, and other factors. So, when the training set and the test set came from different corpus, the training and testing data often follow different feature distributions. The recognition performance will be reduced at this time.

Therefore, transfer learning is adopted to solve the problem of data cross-corpus [28]. The known corpus data is considered as the source domain, and the unknown data to be learned constitutes the target domain. Transfer learning is to transfer the knowledge of the source domain to the target domain to reduce the data distribution difference between the two domains, and in SER, the features of the source and target domains are distributed in different spaces. So, the transfer from the source domain to the target domain is a feature-based transfer, that is, a mapping relationship between two domains is established to reduce the differences in feature distributions. With the development of transfer learning, more transfer learning algorithms are applied to SER. Among them, in order to solve the cross-corpus SER problem, many researches focus on transfer subspace learning and domain adaptation, such as unsupervised transfer subspace learning [28], transfer subspace learning based on feature selection [29], transfer subspace learning based on non-negative matrix factorization [30], transfer linear subspace learning [31], and Universum autoencoder-based domain adaptation [32]. In addition, a cross-corpus speech emotion recognition based on domain adaptive least squares regression is proposed in [33], and in [34, 35], ADDoG-based and DANN-based methods are proposed according to the idea of domain adversarial. Most of the above methods involve transfer subspace learning and domain adaptation, which are important issues in transfer learning and the focus of this paper. The two parts are considered jointly in this paper. Therefore, inspired by the frame in [36], a cross-corpus speech emotion recognition method is proposed.

The contributions of the proposed method are summarized as follows:

- The proposed method combines subspace learning and mapping to realize speech emotion recognition across the corpus. The feasibility of the proposed method is proved by experimental results.
- In this paper, a subspace learning model is constructed based on the Hessian matrix, so that the

extracted features both in the source domain and the target domain have good robustness in their independent subspace, which can be adopted to improve the subsequent cross-corpus transfer ability.

- Information entropy is used to establish a domain adaption model in the proposed method. The numerical descent is used to minimize information entropy, so that a common space of source and target domains is learned, thereby the difference in features distribution between the two domains is reduced.

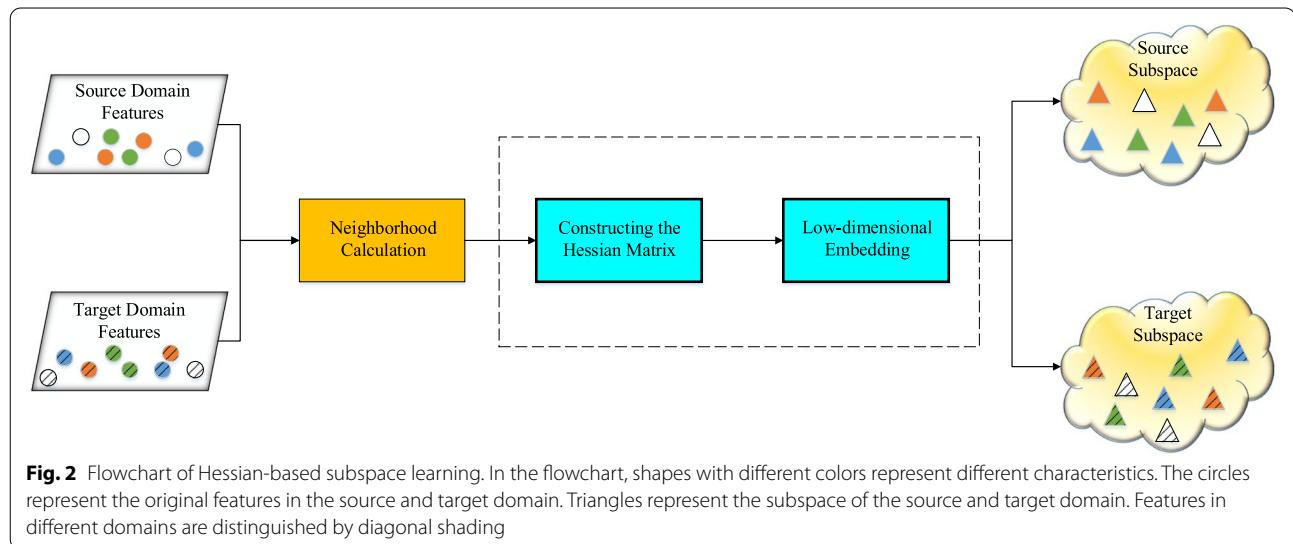
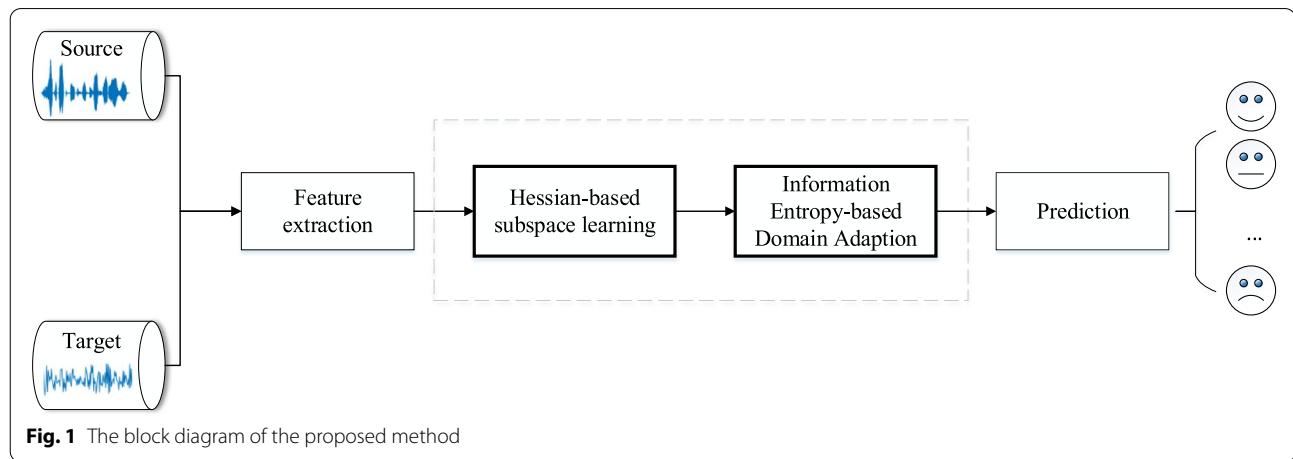
The rest of the paper is organized as follows. In Section 2, the specific process of the proposed method is introduced, along with some optimizations. In Section 3, the emotion recognition performance of the proposed method is analyzed on three public datasets, and the

effects of different parameters on the performance are analyzed through experiments. Finally, the conclusion is drawn in Section 4.

## 2 The proposed method

A cross-corpus speech emotion recognition method is proposed by combing subspace learning and domain adaption. The block diagram of the proposed method is shown in Fig. 1.

Firstly, features of speech in the source corpus and target corpus are extracted to form the source domain and the target domain. Then, the Hessian-based subspace learning is performed on the feature in the source domain and the target domain to obtain low-dimensional features for forming their own independent subspace. The flowchart of the Hessian-based subspace learning part is shown in Fig. 2. Furthermore,



the mapping relationship between the source domain subspace and the target domain subspace is established by using information entropy, which is used for reducing the difference of feature distribution between different domains. This mapping relationship is revealed by the common space. Therefore, it is important to find the common space corresponding to the two domains in this method. The flowchart of the domain adaption part is shown in Fig. 3. Finally, emotions are predicted.

In the part of Hessian-based subspace learning, the neighboring frames of the current frame are found based on neighborhood calculation. Then, the Hessian matrix [37] is constructed for low-dimensional embedding to obtain the subspace of the source and target domain, respectively.

After obtaining the subspace of the source and target domain, the transformation matrix is obtained through correlation coefficients of the subspace. Then, the distance between the feature data of each frame in the source domain subspace with that of each frame in the target domain subspace is calculated. And the probability that a frame in the subspace of the target domain is neighborhood to each frame in the source domain is obtained according to the distance. In this way, the posterior probability that the features of each frame in the target domain subspace are estimated to be a certain class can be obtained according to the known class labels of the features of each frame in the source domain subspace. Then, the entropy between the target domain features and emotion labels and the entropy between the features and domain labels of the two domains are calculated. Finally, the two information entropies are jointly optimized by numerical descent. The mapping relationship between the source domain subspace

and the target domain subspace is acquired, which is described by a common space.

Then, Hessian-based subspace learning [38] and the domain adaption based on information entropy are introduced in detail. Finally, a specific optimization method for finding the common space is given.

## 2.1 Hessian-based subspace learning

An input feature matrix  $\mathbf{X}=(x_{mn})_{M \times N}$  is given, which is composed of the features of the speech.  $m$  and  $n$  are the feature index and the frame index, respectively.  $M$  and  $N$  are the total number of the feature dimension and the number of frames, respectively. First, the feature energy of each frame is as follows:

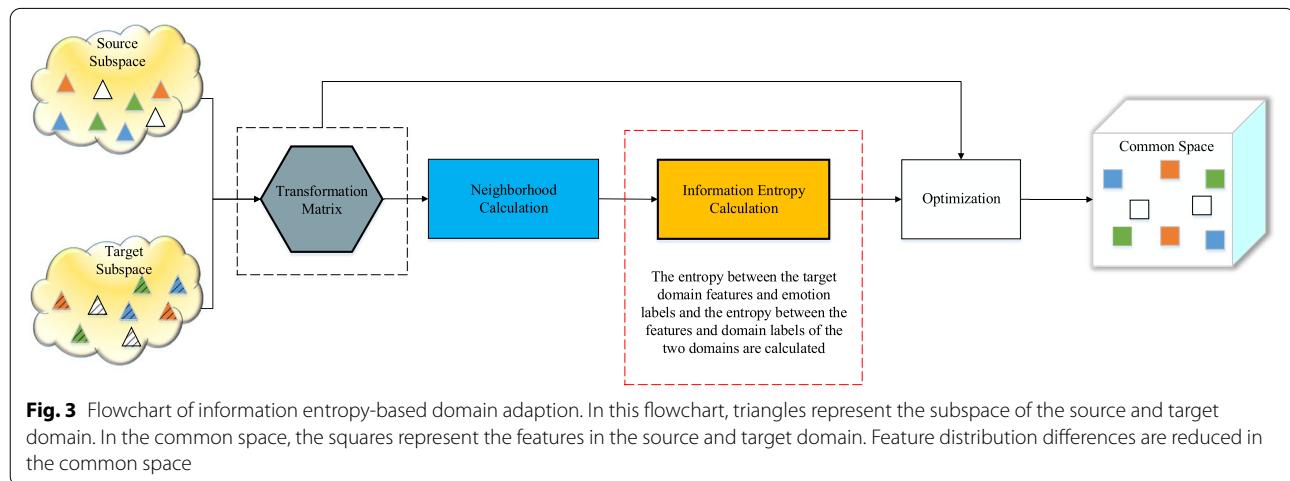
$$x_n^e = \sum_{m=1}^M x_{mn}^2, \quad (1)$$

where  $x_n^e$  represents the feature energy of the  $n$ th frame, and  $x_{mn}$  represents the feature of the  $m$ th dimension in the  $n$ th frame.

Thus, an energy matrix can be formed as  $\mathbf{X}^e = [x_1^e, x_2^e, \dots, x_N^e]$ . Then, two new feature energy matrices  $\mathbf{A}$  and  $\mathbf{B}$ , which are used for calculating the distance of the feature between different frames, are defined as follows:

$$\begin{cases} \mathbf{A} = (a_{ij})_{N \times N} \\ \mathbf{B} = (b_{ij})_{N \times N} \end{cases} \quad (2)$$

where  $a_{ij} = x_j^e$ ,  $b_{ij} = x_i^e$ ,  $1 \leq i, j \leq N$ , and  $i$  and  $j$  represent the index of the row and column, respectively. In order to find the nearest  $K$  frames of each frame, the distance  $\mathbf{D}_e = (d_{ij})_{N \times N}$  of the feature between different frames is calculated as follows:



$$\mathbf{D}_e = \mathbf{A} + \mathbf{B} - 2\mathbf{X}^T \mathbf{X} \quad (3)$$

where  $d_{ij}$  represents the distance between the feature energy of the  $i$ th frame and the  $j$ th frame. The smaller the distance  $d_{ij}$  is, the closer the feature energies of the  $i$ th frame and the  $j$ th frame are. In fact, the definition of distance  $\mathbf{D}_e$  is derived from Euclidean distance.  $\mathbf{A}$  and  $\mathbf{B}$  are formed by the square of the elements in the input matrix  $\mathbf{X}$ . According to Eqs. (1), (2), and (3), the distance defined in this paper meets the requirements of non-negativity, directness, and identity.  $\mathbf{A}$  and  $\mathbf{B}$  are constructed in a way that also satisfies the symmetry of the distance.

The  $j$ th column from the matrix  $\mathbf{D}_e$  (i.e.,  $\mathbf{d}_j^e = [d_{1j}^e, d_{2j}^e, \dots, d_{Nj}^e]^T$ ) denotes the distance vector of feature energy between the  $j$ th frame and each frame. The sorted distance matrix in ascending order is  $\mathbf{d}_j^{es} = [d_{S_j(1)j}^e, d_{S_j(2)j}^e, \dots, d_{S_j(N)j}^e]^T$ ;  $S_j(i)$  denotes the index of the frame sorted by the distance from the  $j$ th frame, where  $S_j(1)$  represents the index with the minimum distance in  $d_{ij}^e$ ; and  $S_j(N)$  is the index of the maximum distance. It is worth mentioning that for each frame,  $d_{jj}^e$  is the minimum element in  $\mathbf{d}_j^e$ , i.e.,  $S_j(1)=j$ . The 2nd to the  $(K+1)$ -th minimum distance from  $\mathbf{d}_j^{es}$  are selected to form the adjacent index matrix  $\mathbf{i}_j = [S_j(2), S_j(3), \dots, S_j(K+1)]^T$  of the  $j$ th frame.  $K$  denotes the number of the largest neighbor frames. Thereby, the  $K \times N$  adjacent index matrix  $\mathbf{I} = [\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_N]$  of  $N$  frames is obtained. Then, the elements in the input matrix  $\mathbf{X}$  correspond to the indices in  $\mathbf{I}$  and are selected to form a neighborhood matrix  $\mathbf{Z}_n$  which is defined as follows:

$$\mathbf{Z}_n = (z_{mk}^n)_{M \times K} \quad (4)$$

where  $z_{mk}^n = x_{mS_n(k+1)}$ ,  $1 \leq k \leq K$ ,  $1 \leq m \leq M$ ,  $1 \leq n \leq N$ .  $k$ ,  $m$ , and  $n$  are the neighbor index, the feature index, and the frame index, respectively.  $\mathbf{Z}_n$  represents the neighborhood matrix corresponding to the  $n$ th frame.

$\mathbf{E}_n$  is a centralized matrix of  $\mathbf{Z}_n$  which is defined as follows:

$$\mathbf{E}_n = (e_{mk}^n)_{M \times K} \quad (5)$$

$$\text{where } e_{mk}^n = \frac{1}{K} \sum_{k=1}^K z_{mk}^n$$

The purpose of the proposed Hessian-based subspace learning is to obtain the local coordinates of the neighborhood, which are transitioned by tangent coordinates. The tangent space consists of tangent coordinates, which is regarded as a subspace of the Euclidean space. A standard orthogonal coordinate

system is associated with the inner product inheritance of the Euclidean space, which can be obtained by using singular value decomposition. Therefore,  $\mathbf{Z}_n - \mathbf{E}_n$  is subjected to singular value decomposition. The standard orthonormal basis  $\mathbf{V}_n = (v_{ij}^n)_{K \times K}$  can be obtained by singular value decomposition as follows:

$$\mathbf{Z}_n - \mathbf{E}_n = \mathbf{U}_n \Sigma_n \mathbf{V}_n^T \quad (6)$$

where  $(\cdot)^T$  denotes transposition.  $\mathbf{U}_n$  is the left singular vector of  $\mathbf{Z}_n - \mathbf{E}_n$ ,  $\Sigma_n$  is a diagonal matrix of singular values.

First  $d$  columns of  $\mathbf{V}_n$  are extracted to constitute the tangent coordinates  $\mathbf{V}_n^d = (v_{ij}^n)_{K \times d}$  with dimension  $K \times d$ .

Next, an association Hessian matrix  $\mathbf{Q}_n$  is given by using  $\mathbf{V}_n^d$ , which is defined as follows:

$$\mathbf{Q}_n = (q_{kj}^n)_{K \times \frac{d(d+1)}{2}} \quad (7)$$

where  $q_{kj}^n = v_{kj_1}^n v_{kj_2}^n$ ,  $n$  is the frame index,  $1 \leq n \leq N$ .  $j_1$  and  $j_2$  are the dimension indexes. The corresponding relationship among  $j$ ,  $j_1$ , and  $j_2$  is given as follows:

$$j = j_2 + \sum_{l=1}^{j_1-1} \sum_{i=j_1}^d 1 \quad (8)$$

$$\text{where } 1 \leq j_1 \leq d, 1 \leq j_2 \leq d, j = 1, 2, \dots, \frac{d(d+1)}{2}.$$

$$\text{Furthermore, an estimation matrix } \mathbf{L}_n = (l_{ij}^n)_{K \times \left(1+d+\frac{d(d+1)}{2}\right)}$$

is constructed as follows:

$$l_{ij}^n = \begin{cases} \frac{1}{v_{ij}^n} & j = 1 \\ q_{ij}^n & 2 \leq j \leq d \\ d+1 & d+1 \leq j \leq \frac{d(d+1)}{2} \end{cases} \quad (9)$$

$$\text{where } 1 \leq i \leq K, 1 \leq n \leq N.$$

$$\mathbf{G}_n = (g_{ij}^n)_{K \times \left(1+d+\frac{d(d+1)}{2}\right)}$$

can be obtained by Schmitt orthogonalization of estimated matrix  $\mathbf{L}_n$  [39]. The last  $\frac{d(d+1)}{2}$  columns of  $\mathbf{G}_n$  are taken to obtain the matrix  $\mathbf{G}_n^b = (g_{ij}^{bn})_{K \times \frac{d(d+1)}{2}}$ . Then, Hessian quadratic matrix  $\mathbf{H}$  can be constructed by using the matrix  $\mathbf{G}_n^b$ , which is formed as follows:

$$\mathbf{H} = \sum_{n=1}^N \mathbf{C}_n^T \mathbf{C}_n \quad (10)$$

where  $\mathbf{C}_n = (c_{ij})_{\frac{d(d+1)}{2} \times N}$  is a matrix composed of  $\mathbf{G}_n^{bT}$ , and it is defined as follows:

$$c_{iS_n(j)} = \begin{cases} g_{ij}^{bn}, & 1 \leq j \leq K \\ 0, & K < j \leq N \end{cases} \quad (11)$$

where  $1 \leq i \leq \frac{d(d+1)}{2}$ , and  $S_n(j)$  denotes the index of the frame sorted by the distance from the  $n$ th frame,  $1 \leq n \leq N$ .

Next, the  $d$ -dimensional subspace corresponding to the  $d$  smallest eigenvalues can be obtained by using  $\mathbf{H}$ , which is a null space and denotes as  $\mathbf{U} = (u_{ij})_{N \times d}$ . If a manifold is locally equidistant to an open subset in Euclidean space, then the mapping function from this manifold to the open subset is a linear function. The quadratic mixed derivative of the linear function is 0, so the local quadratic form formed by the Hessian coefficients is also 0. Hence, the global Hessian matrix has a  $(d+1)$ -dimensional null space. The first-dimension subspace of the Hessian matrix is composed of a constant function, and other  $d$ -dimensional subspaces form equidistant coordinates. Then, the embedding matrix  $\mathbf{R} = (r_{ij})_{d \times d}$  can be calculated as follows:

$$r_{ij} = \sum_{l \in J} u_{li} u_{lj} \quad (12)$$

where  $J$  represents the set of the index of the neighborhood frames,  $1 \leq i \leq d$ ,  $1 \leq j \leq d$ .

Finally, the subspace  $\mathbf{Y}$  is obtained according to the low-dimensional embedding:

$$\mathbf{Y} = \mathbf{R}^\mu \mathbf{U}^T \quad (13)$$

where  $\mu$  is a regularization parameter, and  $(\cdot)^T$  denotes transposition.

There may be a small number of outliers in the subspace  $\mathbf{Y}$  after the low-dimensional embedding. In order to solve this problem, the outliers in the subspace  $\mathbf{Y}$  are corrected in this paper. These outliers are characterized by a small number, with values that deviate from the distribution of most data. So, the detection thresholds are set to recognize the outliers. Then, the outliers are replaced with  $2Tr(\mathbf{U}^T \mathbf{E} \mathbf{U})$  [40], where  $Tr(\cdot)$  means the trace of the matrix in parentheses.  $\mathbf{E} = (e_{ij})_{N \times N}$  is a diagonal matrix, where  $e_{ij}$  is defined as [41]:

$$e_{ij} = \begin{cases} \frac{1}{2\|u_i\|_2} & i = j \\ 0 & i \neq j \end{cases} \quad (14)$$

Following the above steps, the source domain subspace  $\mathbf{Y}_s$  and the target domain subspace  $\mathbf{Y}_t$  can be obtained.

## 2.2 Information entropy-based domain adaption

A domain adaption method was proposed to build the relationship between the source domain subspace

and the target domain subspace. In detail, a common space with similar feature distributions in the source and target domains is constructed. Both the information entropy between the data and emotion labels and the entropy between data and domain labels are used to optimize the mapping [42]. Thereby, the difference in feature distribution in different corpora can be reduced.

After obtaining the source domain subspace  $\mathbf{Y}_s = (y_{ij}^s)_{d \times N}$  and target domain subspace  $\mathbf{Y}_t = (y_{ij}^t)_{d \times N}$ , a principal component coefficient of the source domain  $\mathbf{W}_s = (w_{ij}^s)_{d \times d}$  and the target domain  $\mathbf{W}_t = (w_{ij}^t)_{d \times d}$  is calculated. In some cases, the dimension of the source domain and the target domain is different, which leads to different dimensions of the principal component coefficients. The dimension of the principal component coefficient of the target domain and the source domain with the smallest dimension should be taken as  $d_w$ . The dimensions of the source domain and the target domain are the same in this paper, so  $d_w$  is set as  $d$ . Since the transfer is carried out from the source domain to the target domain, the target domain is used as the basis for the transformation space. The transformation matrix  $\mathbf{W}$  for both source domain and target domain is set as  $\mathbf{W} = \mathbf{W}_t$ . Features in the source domain and target domain can be mapped into a common space by  $\mathbf{W}$ .

First, the distance matrix  $\mathbf{D} = (d_{ij})_{N \times N}$  formed by the features between different frames from the source domain subspace and the target domain subspace is given as follows:

$$\mathbf{D} = \mathbf{A}' + \mathbf{B}' - 2\mathbf{X}_s^T \mathbf{X}_t \quad (15)$$

where  $\mathbf{X}_s = (x_{mn}^s)_{d \times N} = \mathbf{W}^T \mathbf{Y}_s$  denotes the source domain subspace features in transform space,  $\mathbf{X}_t = (x_{mn}^t)_{d \times N} = \mathbf{W}^T \mathbf{Y}_t$  denotes the target domain subspace features in transform space,  $\mathbf{A}' = (a_{ij})_{N \times N}$ ,  $a_{ij} = \sum_{m=1}^d (x_{mj}^s)^2$ ,  $\mathbf{B}' = (b_{ij})_{N \times N}$ ,  $b_{ij} = \sum_{m=1}^d (x_{mi}^t)^2$ .

The neighbor frames are detected according to the distance between the feature of each frame. Therefore, a conditional probability model is defined as follows:

$$p_{ij} = \frac{e^{-d_{ij}}}{\sum_{i=1}^N e^{-d_{ij}}} \quad (16)$$

where  $1 \leq i \leq N$ ,  $1 \leq j \leq N$ , and  $p_{ij}$  is the conditional probability density that the  $j$ th frame in the target domain is adjacent to the  $i$ th frame in the source domain. It can describe the probability of the nearest neighbor between each frame feature in the source domain and the frame feature in the target domain.

The emotion label corresponding to the  $i$ th frame in the source domain is  $\text{Label}_i$ ,  $\text{Label}_i \in \text{Label} = \{1, 2, \dots, L\}$ , i.e., there are a total of  $L$  types of emotion. According to formula (16), an emotion label probability estimate  $\hat{p}_{lj}$  of the  $j$ th frame in the target domain is given as follows:

$$\hat{p}_{lj} = \sum_{\text{Label}_i=l} p_{ij} \quad (17)$$

where  $1 \leq l \leq L$ ,  $1 \leq j \leq N$ ,  $1 \leq i \leq N$ , and  $\hat{p}_{lj}$  express the probability that the  $j$ th frame in the target domain is discriminated as the  $l$ th type of emotion when the emotion of the source domain is known.

Since  $\hat{p}_{lj}$  is a preliminary probability estimate of the emotion label of each frame feature in the target domain, the relationship between target domain features and emotion labels cannot be directly revealed by  $\hat{p}_{lj}$  [43–45]. Therefore, the entropy  $I(\mathbf{X}_t; \text{Label})$  between the target domain features and emotion labels is calculated by using  $\hat{p}_{lj}$  in this paper, which is defined as follows:

$$I(\mathbf{X}_t; \text{Label}) = - \sum_{l=1}^L \left( \log \left( \sum_{j=1}^N \frac{\hat{p}_{lj}}{N} \right) \sum_{j=1}^N \frac{\hat{p}_{lj}}{N} \right) - \frac{\left( -\sum_{j=1}^N \sum_{l=1}^L (\hat{p}_{lj} \log (\hat{p}_{lj})) \right)}{N} \quad (18)$$

Equation (18) is composed of two parts. In the first part, the entropy of the average probability that the feature of all frames in the target domain belongs to each emotion label is calculated. The average of the entropy of the feature in the target domain belonging to each emotion label is computed in the second part. In order to reduce the influence of incorrect labels on the feature discrimination results of each frame in the target domain, Eq. (18) needs to be optimized later. It should be noted that if only the second part is minimized, a degenerate solution will be obtained. That is, all frames in the target domain may be classified into the same type of emotion. So, the first part in Eq. (18) is necessary.

Then, the entropy  $I^{st}(\mathbf{X})$  between the features and domain labels of the two domains are introduced to maximize the similarity between the two domains, which is defined as:

$$I^{st}(\mathbf{X}) = - \sum_{t=1}^2 \left( \sum_{j=1}^{N+M} \frac{p_{tj}}{N+M} \log \left( \sum_{j=1}^{N+M} \frac{p_{tj}}{N+M} \right) \right) - \frac{\left( -\sum_{j=1}^{N+M} \sum_{t=1}^2 (p_{tj} \log (p_{tj})) \right)}{N+M} \quad (19)$$

where  $1 \leq j \leq N+M$ .

To calculate the entropy  $I^{st}(\mathbf{X})$ , firstly, the distance  $d'_{ij}$  between the  $i$ th frame feature in the source domain and the  $j$ th frame feature in the target domains is calculated according to Eq. (3), where  $\mathbf{X} = (x_{ij})_{d \times (N+M)}$  denotes the feature for all frames in the source and target

domains,  $\mathbf{A} = (a_{ij})_{(N+M) \times (N+M)}$ ,  $a_{ij} = \sum_{m=1}^d (x_{mj})^2$ ,  $\mathbf{B} = (b_{ij})_{(N+M) \times (N+M)}$ , and  $b_{ij} = \sum_{m=1}^d (x_{mi})^2$ .  $N$  and  $M$  denote the number of frames in the source domain and target domain, respectively. In this paper, the number of frames in the source domain is the same as that in the target domains, i.e.,  $N = M$ . Then, the probability  $p'_{ij}$  of the  $i$ th frame feature and the  $j$ th frame being adjacent to each other in the source domain and the target domain is calculated according to Eq. (16) using  $d'_{ij}$ . Next, the probability  $p_{tj}$  that the  $j$ th frame in the source domain and the target domain is judged as the target domain or the source domain is calculated according to Eq. (17).

### 2.3 Optimization

In this subsection, an iterative optimization algorithm based on numerical descent [46] is introduced using Eqs. (18) and (19). The objective function is:

$$f = \min \left\{ \lambda I^{st}(\mathbf{X}) - I(\mathbf{X}_t; \text{Label}) \right\} \quad (20)$$

where  $\lambda$  is the regularization parameter.

In the optimization process, the transfer coefficient matrix  $\mathbf{g}$  is given for numerical descent in this paper, which is defined as follows:

$$\mathbf{g} = \lambda \mathbf{g}^{st}(\mathbf{X}) - \mathbf{g}(\mathbf{X}_t; \text{Label}) \quad (21)$$

where  $\lambda$  is the regularization parameter.

The calculation process of  $\mathbf{g}(\mathbf{X}_t; \text{Label})$  is as follows. First, an information matrix  $\mathbf{I}^c = (i_{lj}^c)_{L \times N}$  is defined using  $\hat{p}_{lj}$  as:

$$i_{lj}^c = \frac{\log (\hat{p}_{lj}) - \log \left( \sum_{j=1}^N \frac{\hat{p}_{lj}}{N} \right)}{N} \quad (22)$$

where  $i_{lj}^c$  represents the difference between the probability that the feature of the  $j$ th frame in the target domain belongs to the emotion of the  $l$ th category and the average probability that the features of all frames in the target domain belong to the emotion of the category.

---

**Input:**  $\mathbf{g}$  ► Transfer parameter for the first iteration  
**Input:**  $f$  ► Information entropy for the first iteration  
**Input:**  $L_{step}$  ► Stepsize as a function of the number of iterations  
**Input:**  $L_{maxstep}$  ► Maximum stepsize  
**Input:**  $N_{maxiter}$  ► The maximum number of iterations  
**Input:**  $\mathbf{W}$  ► Transformation matrix  
**Input:**  $d$  ► Dimension of the transformation matrix  
**Output:**  $\mathbf{L}$  ► Common space of source and target domains

1. Initialize  $i_{iter}$  ►  $i_{iter}$  is the index of the iteration
2. Initialize  $L_{step}$ ,  $L_{maxstep}$ ,  $N_{maxiter}$
3. for  $i_{iter} = 2$ :  $N_{maxiter}$ 
  - if  $f^{(i_{iter})} > f^{(i_{iter}-1)}$  ►  $f^{(i_{iter})}$  is the Information entropy in  $(i_{iter})$ th iteration.  
 $L_{step}=L_{step} \times c, (0 < c < 1)$
  - else if  
 $L_{step}=L_{step} \times c, (c > 1)$
  - end
  - if  $L_{step} \geq N_{maxiter}$   
 $L_{step}=N_{maxiter}$
  - end
4. Compute the transformation matrix:  $\mathbf{W}_d = \mathbf{W} - L_{step} * \mathbf{g}$
5. Calculate the common space:  $\mathbf{L} = \sqrt{\frac{d}{\text{trace}(\mathbf{W}_d^T \mathbf{W}_d)}} \mathbf{W}_d$  ►  $\text{trace}(\cdot)$  means to trace the matrix in parentheses  
► stop condition:  $\text{Min}\{|f^{(i_{iter}-3)} - f^{(i_{iter}-2)}|, |f^{(i_{iter}-2)} - f^{(i_{iter}-1)}|, |f^{(i_{iter}-1)} - f^{(i_{iter})}|\} > \epsilon \times f^{(i_{iter})}$   
or  $i_{iter} = N_{maxiter}$
6. Recalculate  $f^{(i_{iter}+1)}$  and  $\mathbf{g}$
7. **break**
8. **end for**
9. **return**  $\mathbf{L}$

---

**Algorithm 1.** Optimization method based on numerical and information entropy and calculation method of mapping space

Next, a coefficient matrix  $\Gamma = (\gamma_{ij})_{N \times N}$  is calculated from  $p_{ij}$  and  $i_{lj}^c$  as follow:

$$\gamma_{ij} = \left( \sum_{i=1}^N o_{ij} p_{ij} - o_{ij} \right) p_{ij} \quad (23)$$

where  $o_{ij} = i_{lj}^c$ ,  $\text{Label}_i = l$ .  $\mathbf{g}(\mathbf{X}_t; \text{Label})$  is obtained as follows:

$$\mathbf{g}(\mathbf{X}_t; \text{Label}) = 2[\mathbf{Y}_s \Omega \mathbf{Y}_s^T + \mathbf{Y}_t \Omega \mathbf{Y}_t^T - \mathbf{Y}_s \Gamma \mathbf{Y}_t^T - \mathbf{Y}_t \Gamma \mathbf{Y}_s^T] \mathbf{W} \quad (24)$$

where  $\Omega$  is a diagonal matrix, and the main diagonal element is  $\sum_{j=1}^N \gamma_{ij}$ .  $\mathbf{W}$  is the transfer matrix.

Since the calculation process of  $\mathbf{g}(\mathbf{X}_t; \text{Label})$  and  $\mathbf{g}^{st}(\mathbf{X})$  is the same, the calculation process of  $\mathbf{g}(\mathbf{X}_t; \text{Label})$  is introduced in detail in this paper. The variables for the calculation process of  $\mathbf{g}^{st}(\mathbf{X})$  refer to the calculation process of  $I^t(\mathbf{X})$ .

Finally, the common space  $\mathbf{L}$  is obtained. So, the feature data in the source domain after mapping is  $\mathbf{F}_s = \mathbf{Y}_s^T \mathbf{L}$ , and the feature data from the target domain is  $\mathbf{F}_t = \mathbf{Y}_t^T \mathbf{L}$ .

### 3 Experiments and results analysis

To evaluate the effectiveness of the proposed cross-corpus speech emotion recognition method, a number of experiments are conducted with some baseline methods

on three commonly standard datasets, namely Berlin [47], NNIME [48], IEMOCAP [49], MSP-Improv [50], and MSP-PODCAST [51]. The specific statistics of each dataset are shown in Table 1.

### 3.1 Data preparation

Berlin dataset is a German emotional speech corpus recorded by the Technical University of Berlin. In this dataset, ten actors performed 7 emotions, including neutral, angry, fearful, happy, sad, disgusted, and bored. The sampling rate is 16 kHz. The dataset contains 233 male emotional sentences and 302 female emotional sentences saved in WAV format.

The NTHU-NTUA Chinese Interactive Multimodal Emotional Corpus (i.e., NNIME) is a multimodal dataset. In this dataset, audio, video, ECG, etc. were recorded for 44 actors during oral interactions. There are 6 emotions including anger, happy, sad, neutral, frustration, and surprise in this dataset. The audio sampling rate is 16 kHz. The dataset also contains annotation results from 49 annotators in different perspectives.

IEMOCAP, known as the Interactive Emotional Binary Motion Capture Database, is recorded by the Speech Analysis and Interpretation Laboratory at the University of Southern California. Ten emotions are shown by recording the expressions, movements, and audio of 10 actors in this dataset. Twelve hours of data are contained in this dataset. The audio sampling rate is 16 kHz. Considering the relevance and ambiguity of different types of emotions, 4 typical emotions (angry, neutral, happy, and sad) audio data were selected from the above three datasets in this paper.

MSP-Improv is an improvised multimodal emotional corpus. There are 6 sessions each session is a dyadic interaction between two speakers. Twenty target sentences are consisted in each session. In this corpus, 12 actors (six male and six female) performed 4 emotions, including neutral, angry, happy, and sad. Two actors improvise these emotion-specific situations, leading them to utter contextualized, non-read renditions of sentences that have fixed lexical content and convey different emotions. The sampling rate is 44.1 kHz. MSP-Improv is more natural than other corpora. Hereinafter referred to as MSP-Improv is MSP.

**Table 1** Database statistics

Database	Language	Number of samples	Emotional kinds
Berlin	German	535	7
NNIME	Chinese	4773	6
IEMOCAP	English	10,039	10
MSP-Improv	English	8438	4
MSP-PODCAST	English	104,267	9

MSP-PODCAST, a large and natural emotional corpus. It relies on existing spontaneous recordings obtained from audio-sharing websites. The criterion to select the podcasts is to include only episodes that can be shared to the broader community. In this corpus, the types of emotions and themes are diverse, and the audio quality is very good in this corpus, because segments recorded with poor quality are removed. Segments with SNR values less than 20 dB are discarded. Phone-quality speech are also removed. Therefore, this step also removes segments that do not have significant energy above 4 kHz. Podcasts in the corpus contain 9 emotions, including angry, sad, happy, neutral, fear, surprise, disgust, others, and contempt. However, angry, happy, neutral, and sad are selected in this paper. There are also many real-world corpora like LSSED [52], and so on.

### 3.2 Experimental settings

In this experiment, 5 artificial audio features are used, including static MFCC and their first- and second-order dynamic differences, LPC, log amplitude-frequency characteristics, Philips Fingerprints [53], and spectral entropy. The selected audio features are listed in Table 2.

In the following, the amplitude characteristic of the frequency coefficient is described by log amplitude-frequency characteristics (LAFC).

Considering that different features contribute differently to speech emotion recognition, each feature in the source domain and the target domain is weighted before training. The weights are set by the dimensions of the features in this paper. For MFCC, LPC, LAFC, Philips Fingerprint, and Spectral Entropy, the corresponding weights are  $\beta_1, \beta_2, \beta_3, \beta_4$ , and  $\beta_5$ , respectively.

After subspace learning and domain adaption, the weighted features in the source domain are trained. That is, the features are used to build a training set. Similarly, the weighted features in the target domain are used to build a test set.

In the training process, a constant recognition accuracy threshold  $\alpha$  is set in advance. Next, the test set is divided into two parts of equal amount of data, i.e., test set 1 and

**Table 2** The features used in this paper

Feature	Feature dimensions
Static MFCC	12
First-order dynamic difference of MFCC	12
Second-order dynamic difference of MFCC	12
LPC	12
LAFC	129
Philips fingerprint	1
Spectral entropy	1

test set 2. Test set 1 is used for assist training, and test set 2 is used to optimize the performance of the proposed method. If the recognition accuracy of a certain type of emotion is less than  $\alpha$  in the first training, the features corresponding to the emotion need to be re-trained in the next training. The operations repeated until one of the following conditions is met: (1) the recognition accuracy of all emotions is greater than  $\alpha$ , and (2) the number of the emotion with recognition accuracy less than  $\alpha$  remains unchanged in the two adjacent training.

To evaluate the performance of the proposed method in the cross-corpus condition, the Berlin, NNIME, and IEMOCAP are combined in pairs in this paper. Then, any two datasets are taken as the source domain and the target domain. Therefore, a total of 6 combination cases are designed as follows:

- N-B: NNIME is the source domain dataset, and Berlin is the target domain dataset.
- B-N: Berlin is the source domain dataset, and NNIME is the target domain dataset.
- N-I: NNIME is the source domain dataset, and IEMOCAP is the target domain dataset.
- I-N: IEMOCAP is the source domain dataset, and NNIME is the target domain dataset.
- B-I: Berlin is the source domain dataset, and IEMOCAP is the target domain dataset.
- I-B: IEMOCAP is the source domain dataset, and Berlin is the target domain dataset.

### 3.2.1 Parameter details

Linear SVM is chosen for training and testing. The grid search method is used to optimize the kernel function coefficients of the SVM and the independent terms of the sum function. There are four hyperparameters and five feature weight coefficients in this experiment. The recognition accuracy threshold  $\alpha$  is set to 0.45. It is determined by an informal experiment. According to the dimension of the feature, the weight coefficient  $\beta_1, \beta_2, \beta_3, \beta_4$ , and  $\beta_5$  are set as 0.3, 0.3, 0.3, 0.05, and 0.05, respectively. The complexity of the algorithm is affected by  $K$ . The larger the value of  $K$  is, the higher the algorithm complexity is, and the more features are extracted. So, the range of the neighboring value  $K$  is set as [3, 9]. For the two regularization parameters  $\mu$  and  $\lambda$ , the range is set to  $\{-1/4, -1/3, -1/2, 1, 1/2, 1/3, 1/4\}$  and  $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ , respectively. Considering that embedding regularization parameter  $\mu$  is an exponent, if  $\mu$  is a positive integer, the value will affect the value of the element in  $\mathbf{Y}$ . Nevertheless, if  $\mu$  is a positive or negative fraction, it may affect the value range of the element in  $\mathbf{R}$ . Hence, both integer and fraction can be chosen for  $\mu$ . For regularization parameter  $\lambda$ , it affects the importance of both parts

of two information entropy. For the proposed method, the dimension of the simplified subspace feature is set to 169.

### 3.2.2 Traditional linear baseline

In order to evaluate the performance of the proposed method for cross-corpus speech emotion recognition, on the basis of the above 6 sets of experiments, the proposed method is compared with some related most commonly used and advanced transfer learning methods. The following is an introduction to these baseline methods:

- Principal components analysis (PCA) [54]: A dimensionality reduction method that maps data into a low-dimensional subspace through linear transformation to prevent information loss as much as possible.
- Linear discriminant analysis (LDA) [55]: In this method, the projection direction that maximizes the ratio of the inter-class distance and minimizes the intra-class distance ratio is found. The subsequent classification results are affected while reducing the dimension.
- Kernel spectral regression (KSR) [56–58]: In reproducing kernel Hilbert spaces (RKHS), the problem of learning embedding functions is transformed by SR into a regression problem.
- Geodesic flow kernel (GFK) [59]: The movement of the domain is simulated by integrating an infinite number of subspaces. The changes in geometric and statistical properties from the source domain to the target domain are described by these subspaces.
- Subspace alignment (SA) [60]: SA is a transfer learning algorithm for two subspaces by matching the feature. The core of this method is to seek linear transformation to transform and align for different data.
- Manifold embedded distribution alignment (MEDA) [61]: Taking into account the importance of both conditional and marginal distributions, a domain-invariant classifier is learned via a Grassmann manifold with structural risk minimization.
- Joint distribution adaptation (JDA) [62]: The marginal probability distribution and conditional probability distribution of the source and target domains are adapted to reduce the distribution difference between different domains.
- Transfer component analysis (TCA) [63]: The data in both domains are mapped together into a high-dimensional regenerated kernel Hilbert space. In this space, the distance of data in the source domain and target domain is minimized.
- Balanced distribution adaptation (BDA) [64]: The weights of marginal and conditional distributions are adaptively utilized on the basis of JDA.
- Transfer joint matching (TJM) [65]: The domain variance is reduced by jointly matching features and

**Table 3** Weighted accuracy (%) of different methods in different cases

Case	Subspace learning				Distribution adaptation					Feature selection	The proposed method
	GFK	PCA	LDA	KSR	SA	MEDA	JDA	TCA	BDA		
N-B	35.63%	32.08%	30.63%	37.29%	36.46%	35.00%	36.88%	31.04%	37.29%	37.50%	50.42%
B-N	32.71%	37.29%	34.79%	40.42%	42.92%	38.75%	49.58%	43.04%	51.04%	45.63%	67.08%
N-I	43.13%	40.83%	47.08%	47.92%	37.29%	42.29%	43.54%	57.50%	44.17%	46.04%	61.25%
I-N	33.33%	38.33%	32.29%	37.71%	41.88%	41.86%	44.79%	41.04%	44.58%	45.00%	67.75%
B-I	41.46%	39.58%	46.67%	50.21%	37.71%	47.71%	33.96%	43.33%	38.54%	46.04%	55.83%
I-B	31.88%	38.75%	42.08%	39.58%	41.04%	41.25%	38.96%	36.67%	47.29%	49.17%	46.88%
Average	36.35%	37.81%	38.92%	42.18%	39.55%	41.14%	41.29%	42.10%	43.82%	44.90%	58.20%

**Table 4** Unweighted accuracy (%) of different methods in different cases

Case	Subspace learning				Distribution adaptation					Feature selection	The proposed method
	GFK	PCA	LDA	KSR	SA	MEDA	JDA	TCA	BDA		
N-B	39.1%	33.75%	31.67%	36.67%	37.71%	30.42%	33.33%	32.71%	36.86%	38.75%	48.54%
B-N	35.28%	38.75%	31.46%	40%	44.58%	31.25%	44.79%	43.75%	42.08%	45%	65.28%
N-I	43.75%	41.86%	32.71%	27.92%	37.5%	38.96%	45.42%	50.42%	31.67%	30.63%	53.33%
I-N	31.04%	30.63%	34.38%	36.67%	42.29%	31.25%	38.33%	35%	39.58%	43.88%	64.04%
B-I	42.79%	38.54%	45%	48.96%	38.33%	39.17%	34.42%	45.42%	37.92%	42.92%	52.28%
I-B	32.91%	36.08%	40.86%	34.79%	34.58%	39.58%	36.67%	34.79%	46.25%	45.83%	46.08%
Average	37.47%	36.6%	36.01%	37.5%	39.17%	35.11%	39.33%	40.35%	39.06%	41.17%	54.93%

reweighting instances across domains in a dimensionality reduction process. The new feature representations invariant to both distributional variance and uncorrelated instances are built.

### 3.3 Results analysis

**3.3.1 Comparison with the traditional linear baseline method**  
In this section, the recognition accuracy of the proposed method is compared with that of some traditional linear baseline methods. The result is shown in Tables 3 and 4.

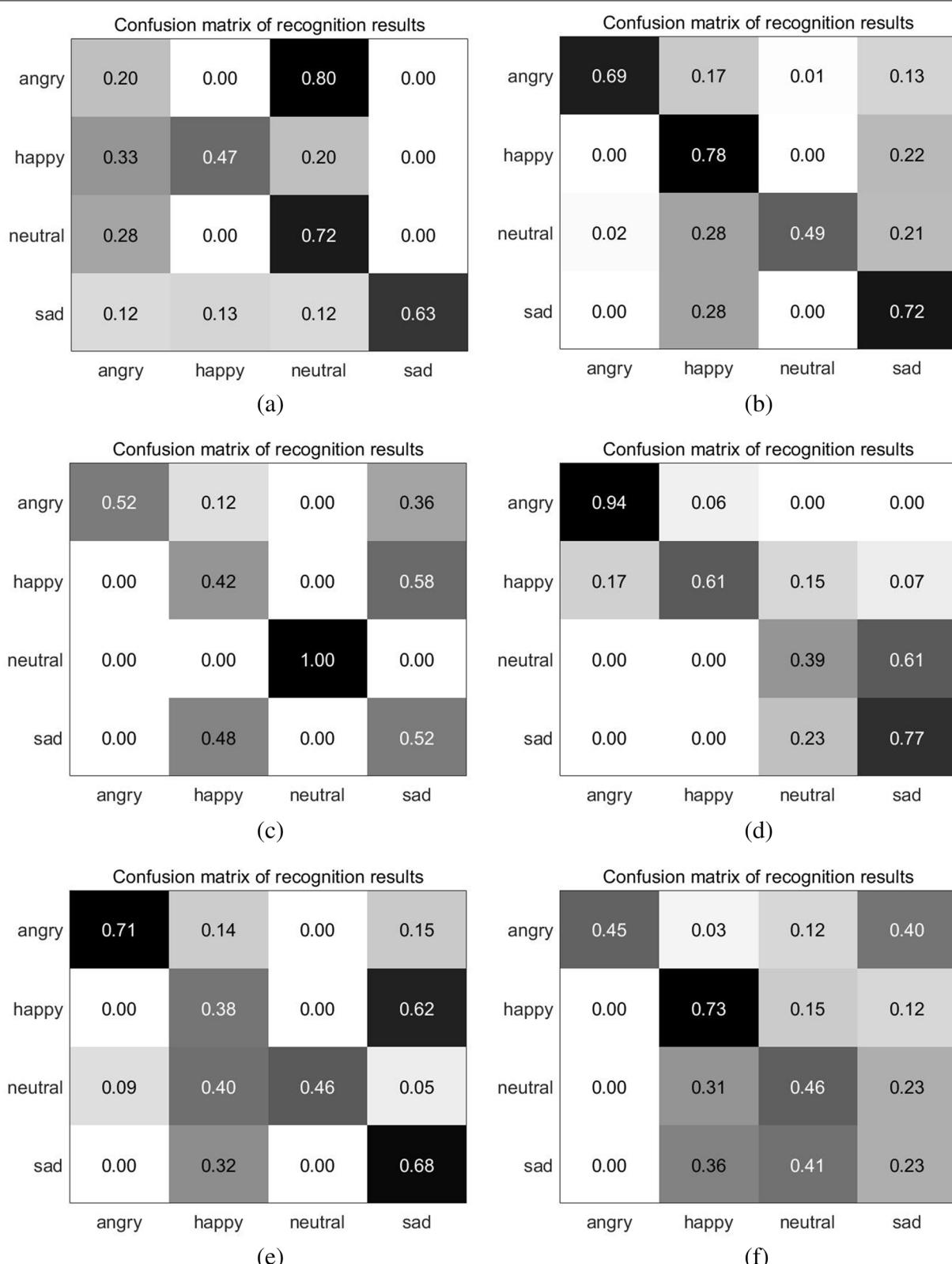
From Table 3, it is clear that the performance of the proposed method outperforms that of other methods in most cases. Only in the case of I-B, the performance of the proposed method is slightly lower than that of BDA and TJM. For the proposed method, the average recognition accuracy reached 58.20% in the six cases. In the case of I-B, the recognition accuracy is the lowest among the six cases, which is 46.88%. In contrast, in the case of I-N, the recognition accuracy reached 67.75%, which is the highest among the six cases. Compared with TJM which has the highest recognition accuracy among the baseline methods, the average recognition accuracy of the proposed method is significantly improved by 13.3%.

Although weighted accuracy is an important indicator to evaluate the overall classification performance of the model, weighted accuracy is affected by the unbalanced distribution of sample classes. Therefore, unweighted

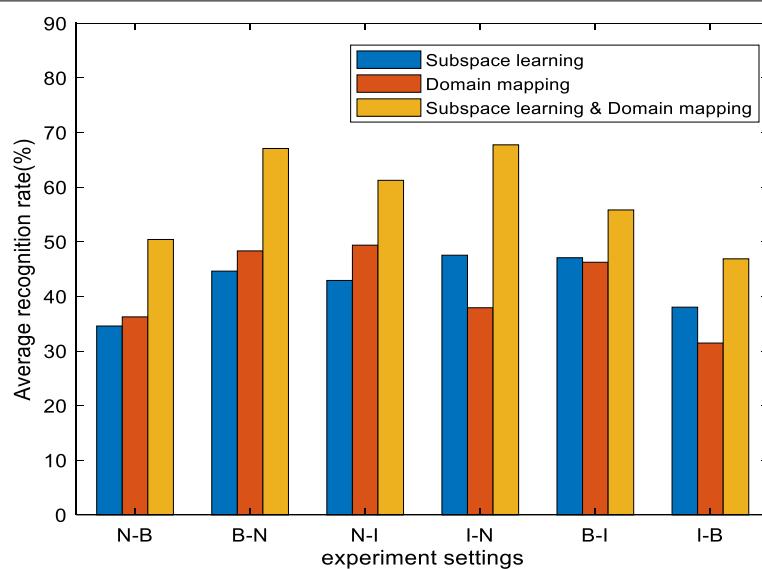
accuracy is very important for evaluating the overall classification performance of the model when the distribution of sample classes is unbalanced. It can be seen from Table 4 that the unweighted accuracy of almost all methods is lower than the weighted accuracy. For the proposed method, unweighted accuracy is 3.27% lower than weighted accuracy. Compared with the baseline method, it still has advantages.

Furthermore, we can find that the average recognition accuracy of the proposed method, distribution adaptation method, and feature selection method is higher than that of most subspace learning. The reason is that the distribution of data in different domains is different. Therefore, the recognition performance of traditional subspace learning algorithms is poor in cross-corpus speech emotion recognition. Transfer learning can be used to improve recognition performance.

In addition, the confusion matrix of the proposed method in six cases is shown in Fig. 4. It can be seen that there are two types of emotion with more than 50% recognition accuracy in most cases. In the case of N-B and N-I, the highest recognition accuracy can be achieved for neutral. From Fig. 4b and f, it is clear that the proposed method has a good recognition ability for happy, and the highest recognition accuracy can be achieved for angry in the case of I-N and B-I. Moreover, it can be also found that sad is easier to be recognized than other emotions in most cases.



**Fig. 4** Confusion matrices for cross-corpus speech emotion recognition of the proposed method in various situations. **a** Confusion matrix in the N-B case. **b** Confusion matrix in the B-N case. **c** Confusion matrix in the N-I case. **d** Confusion matrix in the I-N case. **e** Confusion matrix in the B-I case. **f** Confusion matrix in the I-B case

**Fig. 5** Ablation experiment results

### 3.3.2 Ablation experiment

In this section, a set of ablation experiments is established to verify the impact of the two parts of the proposed method on the recognition performance. The specific results are shown in Fig. 5. The specific settings are as follows:

- Subspace learning: Only Hessian-based Subspace Learning is performed.
- Domain adaption: Only information entropy-based domain adaption is performed.
- Subspace learning and domain adaption: Hessian-based subspace learning and domain adaption are combined.

The average recognition accuracy of the ablation experiments is shown in Fig. 5. It can be found that the recognition performance of the combined method (i.e., the proposed method) is better than that of the method only with Hessian-based subspace learning or domain adaption. Through ablation experiments, it is clear that both Hessian-based subspace learning and domain adaption have played a positive role in cross-corpus speech emotion recognition. In the cases of N-B, B-N, and N-I, the recognition accuracy of the domain adaption method is slightly higher than that of the Hessian-based subspace learning method. On the contrary, in the cases of I-N, B-I, and I-B, the recognition accuracy of the Hessian-based subspace learning method is slightly higher than that of the domain adaption method.

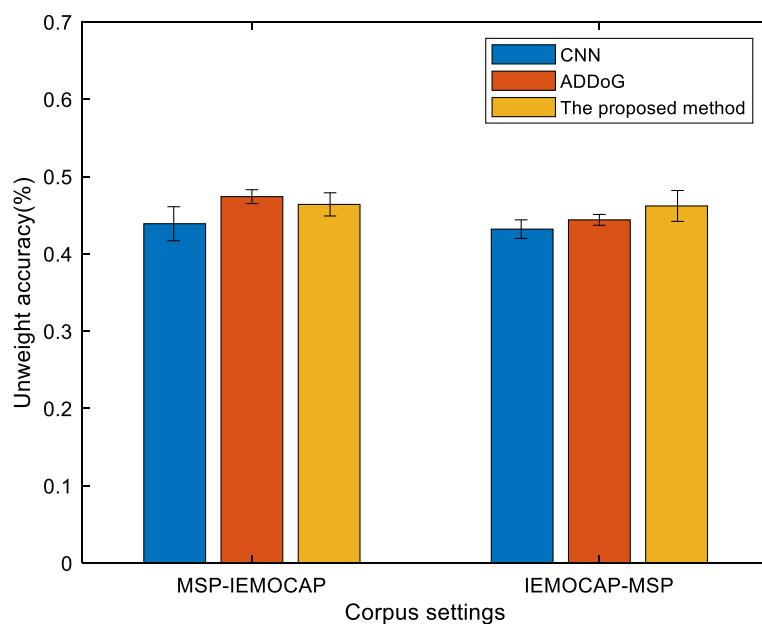
### 3.3.3 Comparison with deep learning-based method

In this section, IEMOCAP and MSP-Improv are used for cross-corpus speech emotion recognition. ADDoG-based method and CNN-based method [34] are chosen as reference methods. The recognition accuracy of the proposed method is compared with these reference methods. The result is shown in Fig. 6.

It can be seen from Fig. 6 that when MSP-Improv is the source domain and IEMOCAP is the target domain, the unweight accuracy of the proposed method is better than that of the CNN-based method but slightly lower than that of the ADDoG-based method. However, in the corpus reverse experiment, the unweight accuracy of the proposed method is slightly higher than that of the CNN-based method and ADDoG-based method. It can be clearly seen that the performance of the ADDoG-based method is the most stable among the three methods. In general, the proposed method can achieve well performance compared with traditional linear methods and deep learning methods.

### 3.3.4 Experiment of real-world corpus

In order to verify that the method proposed in this paper is also effective in the real world, in this section, a real-world corpus MSP-PODCAST and several corpora in controlled experimental environments are used for cross-corpus speech emotion recognition. The experimental setup of this paper is to set MSP-PODCAST as the source corpus and target corpus respectively for experiments with other corpora. The recognition accuracy of the proposed method using MSP-PODCAST as

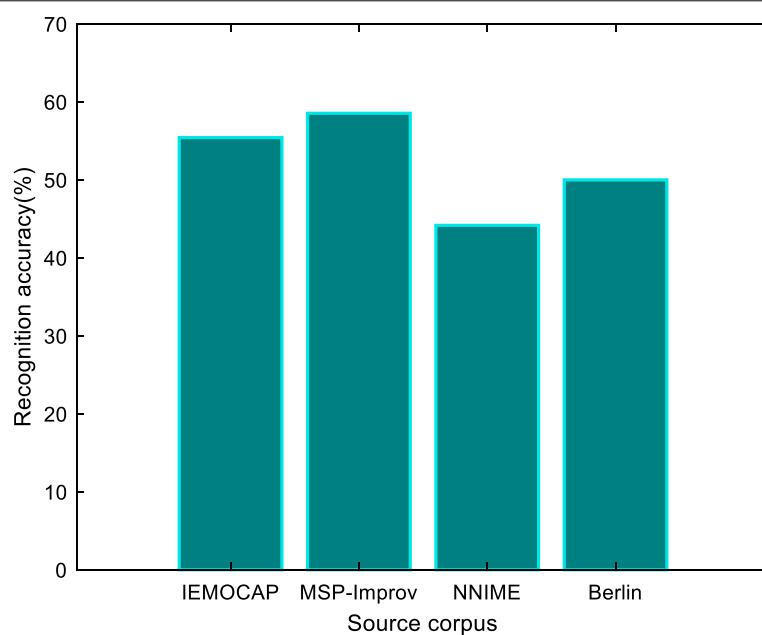


**Fig. 6** Results of the unweight accuracy with IEMOCAP and MSP-Improv

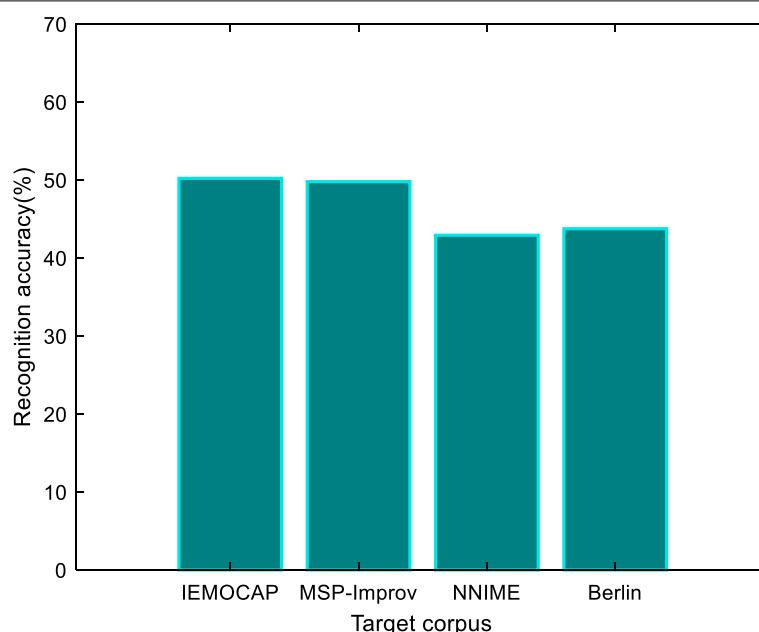
the target corpus is shown in Fig. 7, and Fig. 8 shows the recognition accuracy of the accuracy of MSP-PODCAST as the source corpus:

It can be seen from Figs. 7 and 8 that the recognition performance of the proposed method using MSP-PODCAST as the target corpus is better than that using

MSP-PODCAST as the source corpus. When MSP-PODCAST is used as a source corpus, the transferable knowledge is limited due to the influence of complex acoustic conditions. It can be seen that the performance of speech emotion recognition is indeed affected by the corpus environment. In addition, it is clear that the recognition



**Fig. 7** Recognition accuracy of MSP-PODCAST as target corpus



**Fig. 8** Recognition accuracy of MSP-PODCAST as source corpus

performance of the proposed method using IEMOCAP and MSP-Improv is better than that of other corpora.

### 3.3.5 Parameters analysis

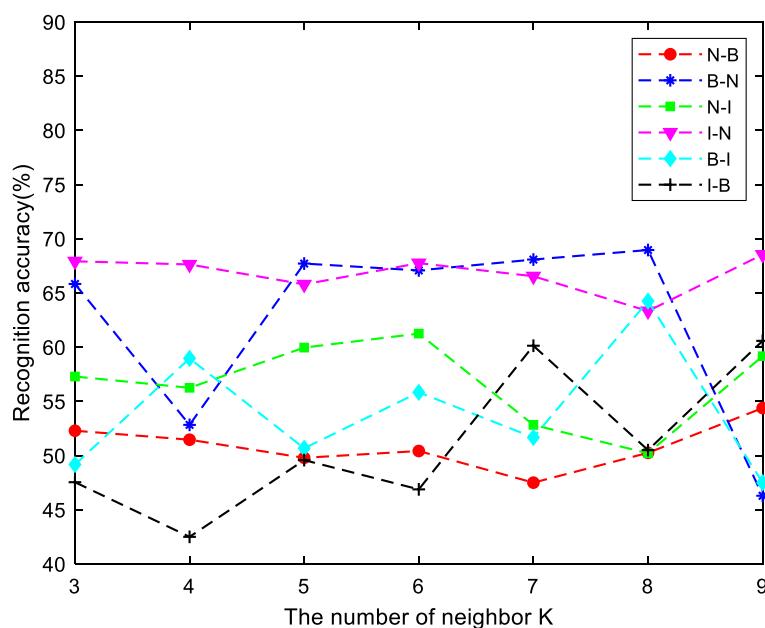
The influence of different parameters on the recognition performance of the proposed method is analyzed in this section. The analyzed parameters include the number of the nearest neighbors  $K$ , the embedding regularization parameter  $\mu$ , and the information entropy regularization parameter  $\lambda$ . Different recognition accuracy can be obtained by selecting different values of parameters.

First of all, the nearest neighbor number  $K$  is analyzed, which is used to identify the number of neighboring frames of the current frame. The complexity of the algorithm is affected by  $K$ . The smaller  $K$  is, the fewer neighboring frames are identified, and the less feature is provided. While the larger  $K$  is, the more neighboring frames are identified, the more feature is provided. However, if  $K$  is set large, some frames which are not useful for recognition may be identified as neighboring frames, which may lead to high algorithmic complexity. So, the range of  $K$  is set from 3 to 9 in this paper. In different cases, the recognition accuracy of different  $K$  is shown in Fig. 9. From Fig. 9, we can find that the proposed method achieves a good recognition accuracy when  $K = 6$ . However, it is not enough to only use the recognition accuracy to measure the recognition performance under different corpus settings. Therefore, variances of recognition

accuracy are introduced in parameter analysis to measure the recognition performance under different corpus settings at the same time in this paper. For  $K$ , variances under different corpus settings are shown in Fig. 10. It can be seen from Fig. 10 that, although the variances of recognition accuracy achieve the maximum when  $K = 6$ , there is a small difference when  $K$  takes different values. Therefore, considering the algorithmic complexity and recognition performance,  $K$  is selected as 6 in this paper.

Then, the embedding regularization parameter  $\mu$  is analyzed, which is used to control the value of the embedded coordinates. The range of  $\mu$  is set as  $\{-1/2, -1/3, -1/4, 1/4, 1/3, 1/2, 1\}$  in this paper. In different cases, the recognition accuracy of the proposed method with different  $\mu$  is shown in Fig. 11. From Fig. 11, it is clear that the proposed method can achieve a good recognition accuracy when  $\mu = 1/4$ . The variance of recognition accuracy with different  $\mu$  under different corpus settings is shown in Fig. 12. Although the variance of recognition accuracy is very small when  $\mu = 1$ , the recognition accuracy is significantly lower than that under other conditions. Therefore, in consideration of recognition accuracy and variance of recognition accuracy,  $\mu = 1/4$  is chosen in this paper.

Finally, the information entropy regularization parameter  $\lambda$  is analyzed, which controls the weight of the information entropy. The range of  $\lambda$  is set as  $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$  in this paper. In different cases, the recognition accuracy of the proposed

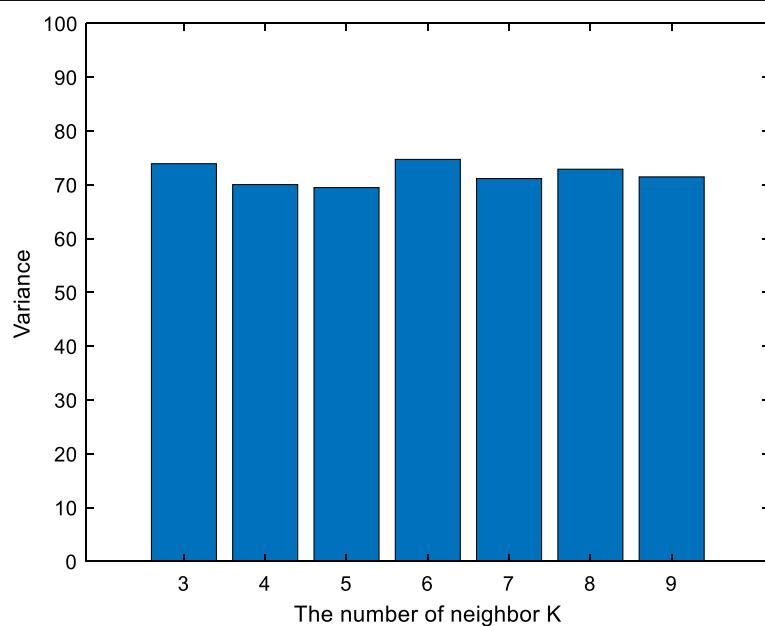
**Fig. 9** Recognition accuracy with different  $K$ 

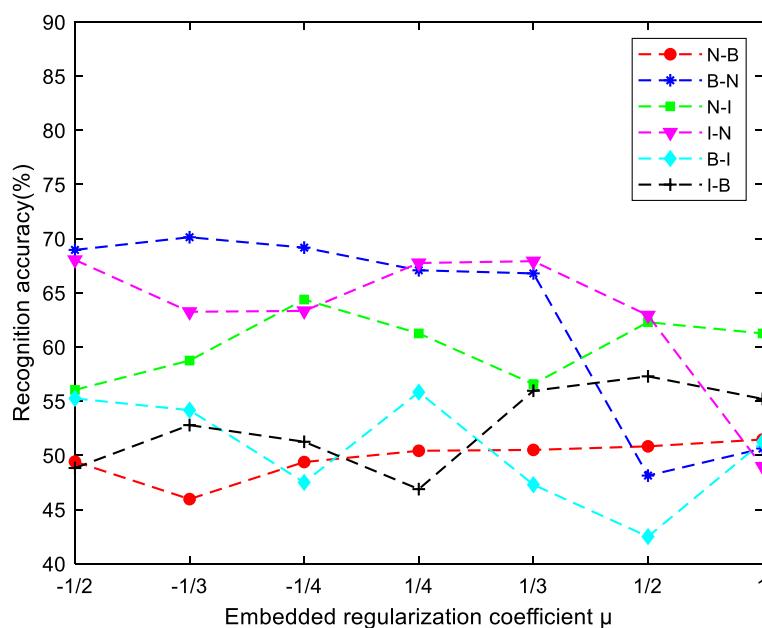
method with different  $\lambda$  is shown in Fig. 13. As shown in Fig. 13, when  $\lambda = 100$  and  $\lambda = 1000$ , the changes in the recognition accuracy are great. Although when  $\lambda = 100$ , the recognition accuracy in both N-I and B-I cases exceeds 70%. However, it is not stable in these two cases as shown in Fig. 14. Therefore, considering recognition accuracy and variance of recognition

accuracy in a compromise,  $\lambda = 10$  is chosen in this paper.

### 3.4 Complexity analysis

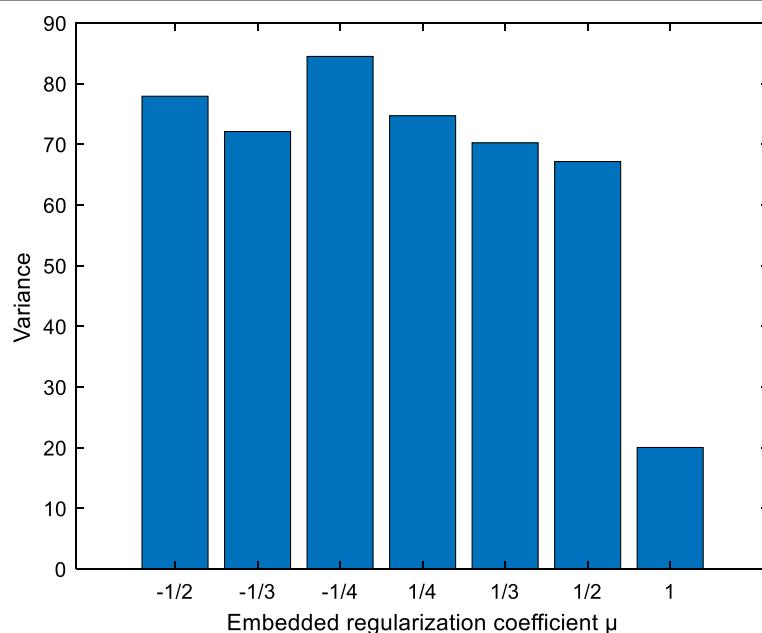
For the performance evaluation of a method, both recognition accuracy and model complexity should be considered.

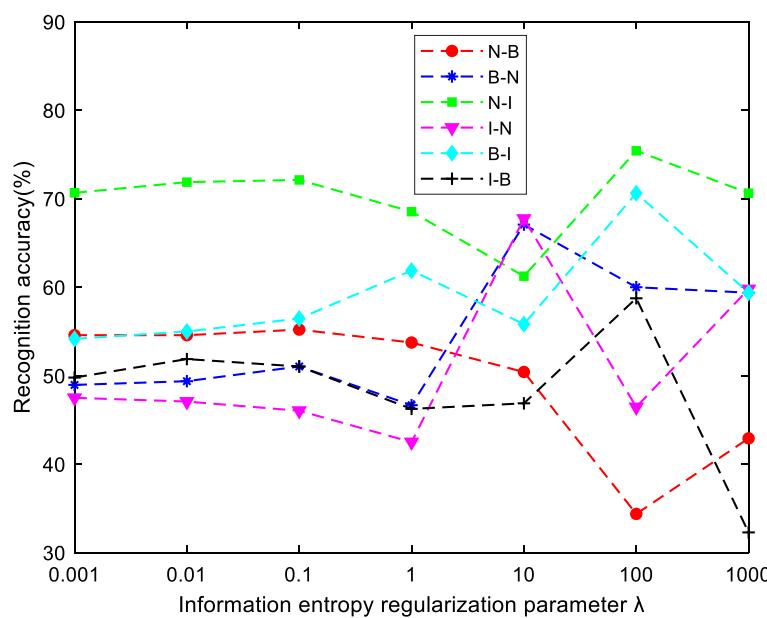
**Fig. 10** Variance of recognition accuracy with different  $K$

**Fig. 11** Recognition accuracy with different  $\mu$ 

For the deep learning-based method, the complexity of the model is determined by the network structure and the number of parameters. Therefore, some complexity analysis of the proposed method and reference methods are given in this subsection. For the CNN-based method, the feature encoder consists of two convolution layers and a max pooling layer, and the emotion classifier consists

of fully connected layers and softmax. On this basis, the ADDoG model adds a critic composed of full connection layers. With the increase of the input MFBs, the calculation amount and trainable parameter amount of each layer will increase more. In addition, during training, when the number of samples in the source domain and target domain increases, the computational complexity of the

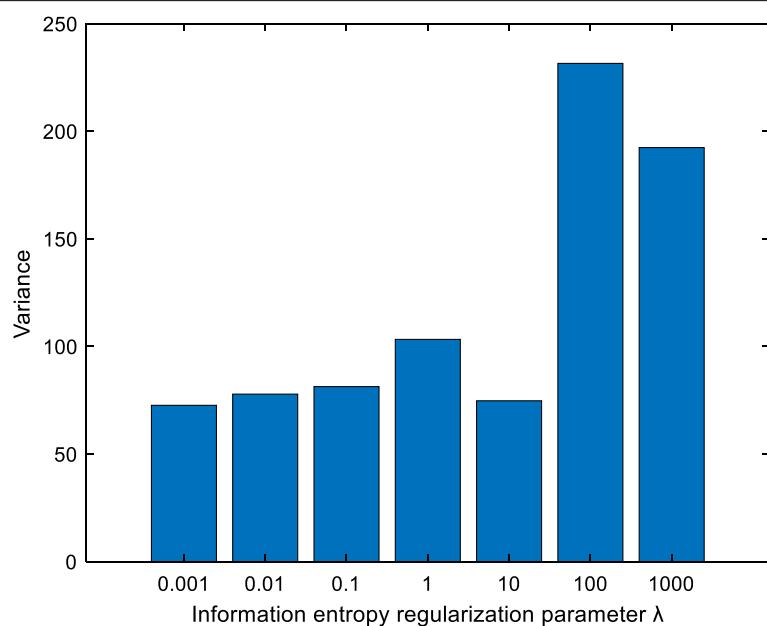
**Fig. 12** Variance of recognition accuracy with different  $\mu$

**Fig. 13** Recognition accuracy with different  $\lambda$ 

loss function and iteration times increase. Although there is a user-defined maximum number of iterations for the proposed method, convergence can be achieved by an average of 50 or fewer iterations under each experimental setting. In summary, the proposed method requires relatively few adaptation steps compared to the needing of fine-tuning whole deep neural network.

#### 4 Conclusion

In this paper, a cross-corpus speech emotion recognition method is proposed using subspace learning and domain adaptation. In the subspace learning part, the Hessian matrix is introduced to locally embed the features in both source and target domains to form the feature subspace. In the domain adaption part, the

**Fig. 14** Variance of recognition accuracy with different  $\lambda$

mapping relationship is constructed based on information entropy. Then, the common space of both the source and target domains is obtained, which reduces the discrepancy in feature distribution between the source and target domains. Extensive experiments on datasets in three different languages are conducted to verify the performance of the proposed method.

#### Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants (61971015), Beijing Natural Science Foundation (No. L223033), and the Cooperative Research Project of BJUT-NTUT (No. NTUT-BJUT-110-05).

#### Authors' contributions

CX performed the whole research and wrote the paper. JM provided support to the writing and experiments. The authors read and approved the final version of the paper.

#### Funding

This work was supported by the National Natural Science Foundation of China under Grants (61971015) and the Cooperative Research Project of BJUT-NTUT (No. NTUT-BJUT-110-05).

#### Availability of data and materials

Not applicable.

#### Declarations

##### Competing interests

The authors declare that they have no competing interests.

##### Author details

<sup>1</sup>Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China. <sup>2</sup>Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan.

Received: 20 August 2022 Accepted: 14 December 2022

Published online: 27 December 2022

#### References

- S. Zhao, G. Jia, J. Yang, G. Ding, K. Keutzer, Emotion recognition from multiple modalities: fundamentals and methodologies. *IEEE Sign. Process. Magazine* **38**(6), 59–73 (2021)
- X. Wu, S. Hu, Z. Wu, X. Liu, H. Meng, in *2022 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP)*. Neural architecture search for speech emotion recognition (2022), pp. 1–4
- C.-C. Lee, K. Sridhar, J.-L. Li, W.-C. Lin, S. Bo-Hao, C. Busso, Deep representation learning for affective speech signal analysis and processing: preventing unwanted signal disparities. *IEEE Sign. Process. Magazine* **38**(6), 22–38 (2021)
- J.S. Gómez-Cañón, E. Cano, T. Eerola, P. Herrera, H. Xiao, Y.-H. Yang, E. Gómez, Music emotion recognition: toward new, robust standards in personalized and context-sensitive applications. *IEEE Sign. Process. Magazine* **38**(6), 106–114 (2021)
- W. Chung-Hsien, W.-B. Liang, Emotion recognition of affective speech based on multiple classifiers using acoustic-prosodic information and semantic labels. *IEEE Trans. Affect. Comput.* **2**(1), 10–21 (2011)
- J.-H. Hsu, M.-H. Su, C.-H. Wu, Y.-H. Chen, Speech emotion recognition considering nonverbal vocalization in affective conversations. *IEEE/ACM Trans. Audio, Speech Lang. Process.* **29**, 1675–1686 (2021)
- B. Chen, Q. Cao, M. Hou, Z. Zhang, G. Lu, D. Zhang, Multimodal emotion recognition with temporal and semantic consistency. *IEEE/ACM Trans. Audio, Speech Lang. Process.* **29**, 3592–3603 (2021)
- B.T. Atmaja, A. Sasou, M. Akagi, Survey on bimodal speech emotion recognition from acoustic and linguistic information fusion. *Speech Commun.* **140**, 11–28 (2022)
- Y. Jin, P. Song, W. Zheng, L. Zhao, Novel feature fusion method for speech emotion recognition based on multiple kernel learning. *J. South. Univ. (English Edition)* **29**(2), 129–133 (2013)
- U. Garg, S. Agarwal, S. Gupta, R. Dutt, D. Singh, in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*. Prediction of emotions from the audio speech signals using MFCC, MEL and Chroma (2020), pp. 87–91
- N.P. Jagini, R.R. Rao, in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*. Exploring emotion specific features for emotion recognition system using PCA approach (2017), pp. 58–62
- S.R. Krishna, R.R. Rao, in *2017 International Conference on Communication and Signal Processing (ICCP)*. Exploring robust spectral features for emotion recognition using statistical approaches (2017), pp. 1838–1843
- J.A. Russell, A circumplex model of affect. *J. Pers. Soc. Psychol.* **39**, 1161–1178 (1980)
- J. Posner, J.A. Russell, B.S. Peterson, The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology. *Dev. Psychopathol.* **17**(3), 715–734 (2005)
- A. Mehrabian, *Basic dimensions for a general psychological theory* (Oelgeschläger, Gunn&Hain, Incorporated, Cambridge, 1980), pp. 39–53
- R.F. Bales, *Social interaction systems: theory and measurement* (Transaction Publishers, Piscataway, 2001), pp. 139–140
- Y. Zhou, X. Liang, Y. Gu, Y. Yin, L. Yao, Multi-classifier interactive learning for ambiguous speech emotion recognition. *IEEE/ACM Trans. Audio, Speech Lang. Process.* **30**, 695–705 (2022)
- Y. Pan, P. Shen, L. Shen, Speech emotion recognition using support vector machine. *Int. J. Smart Home* **6**(2), 101–108 (2012)
- S. Mao, D. Tao, G. Zhang, P.C. Ching, T. Lee, in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.* Revisiting hidden Markov models for speech emotion recognition (2019), pp. 6715–6719
- H. Hu, M. Xu, W. Wu, in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. GMM supervector based SVM with spectral features for speech emotion recognition (2007), pp. IV-413–IV-416
- Y.-C. Kao, C.-T. Li, T.-C. Tai, J.-C. Wang, in *2021 9th International Conference on Orange Technology (ICOT)*. Emotional speech analysis based on convolutional neural networks (2021), pp. 1–4
- C.-H. Park, D.-W. Lee, K.-B. Sim, in *2002 International Conference on Machine Learning and Cybernetics*. Emotion recognition of speech based on RNN, vol 4 (2002), pp. 2210–2213
- S. Wang, X. Ling, F. Zhang, J. Tong, in *2010 International Conference on Measuring Technology and Mechatronics Automation*. Speech emotion recognition based on principal component analysis and back propagation neural network (2010), pp. 437–440
- K.H. Lee, H. Kyun Choi, B.T. Jang, D.H. Kim, in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. A study on speech emotion recognition using a deep neural network (2019), pp. 1162–1165
- X. Wu et al., in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. Speech emotion recognition using sequential capsule networks, vol 29 (2021), pp. 3280–3291
- L. Yi, M.-W. Mak, Improving speech emotion recognition with adversarial data augmentation network. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(1), 172–184 (2022)
- S. Mao, P.C. Ching, T. Lee, Enhancing segment-based speech emotion recognition by iterative self-learning. *IEEE/ACM Trans. Audio, Speech Lang. Process.* **30**, 123–134 (2022)
- N. Liu et al., Transfer subspace learning for unsupervised cross-corpus speech emotion recognition. *IEEE Access* **9**, 95925–95937 (2021)
- P. Song, W. Zheng, Feature selection based transfer subspace learning for speech emotion recognition. *IEEE Trans. Affect. Comput.* **11**(3), 373–382 (2020)
- H. Luo, J. Han, Nonnegative matrix factorization based transfer subspace learning for cross-corpus speech emotion recognition. *IEEE/ACM Trans. Audio, Speech Lang. Process.* **28**, 2047–2060 (2020)
- P. Song, Transfer linear subspace learning for cross-corpus speech emotion recognition. *IEEE Trans. Affect. Comput.* **10**(2), 265–275 (2019)
- J. Deng, X. Xu, Z. Zhang, S. Frühholz, B. Schuller, Universum autoencoder-based domain adaptation for speech emotion recognition. *IEEE Sign. Process. Lett.* **24**(4), 500–504 (2017)
- Y. Zong, W. Zheng, T. Zhang, X. Huang, Cross-corpus speech emotion recognition based on domain-adaptive least-squares regression. *IEEE Sign. Process. Lett.* **23**(5), 585–589 (2016)

34. J. Gideon, M.G. McInnis, E.M. Provost, Improving cross-corpus speech emotion recognition with adversarial discriminative domain generalization (ADDG). *IEEE Trans. Affect. Comput.* **12**(4), 1055–1068 (2021)
35. M. Abdelwahab, C. Busso, Domain adversarial for acoustic emotion recognition. *IEEE/ACM Trans. Audio, Speech Lang. Process.* **26**(12), 2423–2435 (2018)
36. W. Zhang, P. Song, Transfer sparse discriminant subspace learning for cross-corpus speech emotion recognition. *IEEE/ACM Trans. Audio, Speech Lang. Process.* **28**, 307–318 (2020)
37. D.L. Donoho et al., Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. *Proc. Natl. Acad. Sci. U. S. A.* **100**(10), 5591–5596 (2003)
38. Lianbo Zhang, D. Tao and Weifeng Liu, in Proceedings of the 16th International Conference on Communication Technology. Supervised Hessian Eigenmap for dimensionality reduction (IEEE, Hangzhou, China, 2015), pp.903–907.
39. F. Asano, Y. Suzuki, D.C. Swanson, Optimization of control source configuration in active control systems using Gram-Schmidt orthogonalization. *IEEE Trans. Speech Audio Process.* **7**(2), 213–220 (1999)
40. F. Nie, H. Huang, X. Cai, et al, in Proceedings of the 24th Annual Conference on Neural Information Processing Systems. Efficient and Robust Feature Selection via Joint  $\ell_2, 1$ -Norms Minimization (NIPS, Vancouver, BC, Canada, 2010), pp.1–9
41. R. He, T. Tan, L. Wang, W. Zheng, in 2012 IEEE Conference on Computer Vision and Pattern Recognition.  $\ell_2, 1$  regularized correntropy for robust feature selection (2012), pp. 2504–2511
42. Y. Shi, F. Sha, in Proceedings of the 29th International Conference on Machine Learning. Information-Theoretical Learning of Discriminative Clusters for Unsupervised Domain Adaptation (IMLS, Edinburgh, United Kingdom, 2012), pp.1079–1086
43. B. Gholami, P. Sahu, O. Rudovic, K. Bousmalis, V. Pavlovic, Unsupervised multi-target domain adaptation: an information theoretic approach. *IEEE Trans. Image Process.* **29**, 3993–4002 (2020)
44. Y. Tu, M. Mak, J. Chien, Variational domain adversarial learning with mutual information maximization for speaker verification. *IEEE/ACM Trans. Audio, Speech Lang. Process.* **28**, 2013–2024 (2020)
45. D. Xin, T. Komatsu, S. Takamichi, H. Saruwatari, in 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Disentangled speaker and language representations using mutual information minimization and domain adaptation for cross-lingual TTS (2021), pp. 6608–6612
46. X. Wang, L. Yan and Q. Zhang, in Proceedings of the International Conference on Computer Network, Electronic and Automation. Research on the Application of Gradient Descent Algorithm in Machine Learning (IEEE, Xi'an, China, 2021), pp. 11–15
47. F. Burkhardt, A. Paeschke, M. Rolfs, W. Sendlmeier and B. Weiss, in Proceedings of the Interspeech. A database of German emotional speech (ISCA, Lisbon, Portugal, 2005), pp. 1517–1520
48. H.-C. Chou, W.-C. Lin, L.-C. Chang, C.-C. Li, H.-P. Ma, C.-C. Lee, in 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII). NNIME: the NTHU-NTUA Chinese interactive multimodal emotion corpus (2017), pp. 292–298
49. C. Busso, M. Bulut, C.C. Lee, et al., IEMOCAP: interactive emotional dyadic motion capture database. *Lang. Resourc. Eval.* **42**(4), 335–359 (2008)
50. C. Busso, S. Parthasarathy, A. Burmania, M. AbdelWahab, N. Sadoughi, E.M. Provost, MSP-IMPROV: an acted corpus of dyadic interactions to study emotion perception. *IEEE Trans. Affect. Comput.* **8**(1), 119–130 (2017)
51. R. Lotfian, C. Busso, Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings. *IEEE Trans. Affect. Comput.* **10**(4), 471–483 (2019)
52. Fan, Weiquan, et al, in Proceedings of the International Conference on Acoustics, Speech and Signal Processing. LSSED: a large-scale dataset and benchmark for speech emotion recognition (IEEE, Toronto, Canada, 2021), pp. 641–645
53. J. Haitsma, T. Kalker, in Proceedings of the 3rd International Conference on Music Information Retrieval. A highly robust audio fingerprinting system (ISMIR, Paris, France, 2002), pp. 107–115
54. Y.C. Du, W.C. Hu, L.Y. Shyu, in The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. The effect of data reduction by independent component analysis and principal component analysis in hand motion identification (2004), pp. 84–86
55. S. Ji, J. Ye, Generalized linear discriminant analysis: a unified framework and efficient model selection. *IEEE Trans. Neural Netw.* **19**(10), 1768–1782 (2008)
56. D. Cai, Spectral Regression: A Regression Framework for Efficient Regularized Subspace Learning. (Doctoral dissertation, University of Illinois at Urbana-Champaign), 2009
57. D. Cai, X. He, J. Han, Speed up kernel discriminant analysis. *Int. J. Very Large Data Bases* **20**(1), 187–191 (2011)
58. D. Cai, X. He, J. Han, in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. Spectral regression: a unified approach for sparse subspace learning (2007), pp. 73–82
59. B. Gong, Y. Shi, F. Sha, K. Grauman, in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Geodesic flow kernel for unsupervised domain adaptation (2012), pp. 2066–2073
60. B. Fernando, A. Habrard, M. Sebban, T. Tuytelaars, in *2013 IEEE International Conference on Computer Vision*. Unsupervised visual domain adaptation using subspace alignment (2013), pp. 2960–2967
61. J. Wang, W. Feng, Y. Chen, et al, in Proceedings of the ACM Multimedia Conference. Visual Domain Adaptation with Manifold Embedded Distribution Alignment (ACM, Seoul, Korea, 2018), pp. 402–410
62. M. Long, J. Wang, G. Ding, J. Sun, P.S. Yu, in *2013 IEEE International Conference on Computer Vision*. Transfer feature learning with joint distribution adaptation (2013), pp. 2200–2207
63. S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis. *IEEE Trans. Neural Netw.* **22**(2), 199–210 (2011)
64. J. Wang, Y. Chen, S. Hao, W. Feng, Z. Shen, in *2017 IEEE International Conference on Data Mining (ICDM)*. Balanced distribution adaptation for transfer learning (2017), pp. 1129–1134
65. M. Long, J. Wang, G. Ding, J. Sun, P.S. Yu, in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Transfer joint matching for unsupervised domain adaptation (2014), pp. 1410–1417

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen® journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

## Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”). Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

[onlineservice@springernature.com](mailto:onlineservice@springernature.com)

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/328997636>

# Predicting enterprise cyber incidents using social network analysis on the darkweb hacker forums

Preprint · November 2018

CITATIONS  
0

READS  
273

4 authors:



Soumajyoti Sarkar  
Amazon  
31 PUBLICATIONS 115 CITATIONS

[SEE PROFILE](#)



Mohammed Almukaynizi  
Arizona State University  
18 PUBLICATIONS 208 CITATIONS

[SEE PROFILE](#)



Jana Shakarian  
self  
43 PUBLICATIONS 497 CITATIONS

[SEE PROFILE](#)



Paulo Shakarian  
Arizona State University  
187 PUBLICATIONS 2,113 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Strongly Hierarchical Factorization Machines and ANOVA Kernel Regression [View project](#)



Cyber-Attack Prediction [View project](#)

# Predicting enterprise cyber incidents using social network analysis on the darkweb hacker forums

Soumajyoti Sarkar  
*Arizona State University*  
Tempe, USA  
ssarka18@asu.edu

Mohammad Almukaynizi  
*Arizona State University*  
Tempe, USA  
malmukay@asu.edu

Jana Shakarian  
*Cyber Reconnaissance, Inc.*  
Tempe, USA  
jana@cyr3con.ai

Paulo Shakarian  
*Arizona State University*  
Tempe, USA  
shak@asu.edu

**Abstract**—With rise in security breaches over the past few years, there has been an increasing need to mine insights from social media platforms to raise alerts of possible attacks in an attempt to defend conflict during competition. We use information from the darkweb forums by leveraging the reply network structure of user interactions with the goal of predicting enterprise cyber attacks. We use a suite of social network features on top of supervised learning models and validate them on a binary classification problem that attempts to predict whether there would be an attack on any given day for an organization. We conclude from our experiments using information from 53 forums in the darkweb over a span of 12 months to predict real world organization cyber attacks of 2 different security events that analyzing the path structure between groups of users is better than just studying network centralities like Pagerank or relying on the user posting statistics in the forums.

## I. INTRODUCTION

With the recent data breaches such as those of Yahoo, Uber, Equifax<sup>1</sup> among several others that emphasize the increasing financial and social impact of cyber attacks, there has been an enormous requirement for technologies that could provide such organizations with prior alerts on such data breach possibilities. These breaches are a direct or indirect result of cyber, electronic, and information operations to infiltrate systems and infrastructure as well as gain unauthorized access to information, thus setting an example of conflict during competition. On the vulnerability front, the Risk Based Security's VulnDB database<sup>2</sup> published a total of 4,837 vulnerabilities in a quarter of 2017, which was around 30% higher than previous year. This motivates the need for extensive systems that can utilize vulnerability associated information from external sources to raise alerts on such cyber attacks. The darkweb is one such place on the internet where users can share information on software vulnerabilities and ways to exploit them [1], [15]. Surprisingly, it might be difficult to track the actual intention of those users, thus making it necessary to use data mining and learning to identify the discussions among the noise that could potentially raise alerts on attacks on external enterprises.

Some of the authors are supported through the AFOSR Young Investigator Program (YIP) grant FA9550-15-1-0159, ARO grant W911NF-15-1-0282, and the DoD Minerva program grant N00014-16-1-2015.

<sup>1</sup><https://www.consumer.ftc.gov/blog/2017/09/equifax-data-breach-what-do>,  
<https://www.consumer.ftc.gov/blog/2016/09/yahoo-breach-watch>

<sup>2</sup><https://www.riskbasedsecurity.com/2017/05/29/increase-in-vulnerabilities-already-disclosed-in-2017/>

In this paper, we leverage the information obtained from analyzing the reply network structure of discussions in the darkweb forums to understand the extent to which the darkweb information can be useful for predicting real world cyber attacks.

Most of the work on vulnerability discussions on trading exploits in the underground forums [9], [13], [14] and related social media platforms like Twitter [2], [8], [15] have focused on two aspects: (1) analyzing vulnerabilities discussed or traded in the forums and the markets, thereby giving rise to the belief that the "lifecycle of vulnerabilities" in these forums and marketplaces and their exploitation have significant impact on real world cyber attacks [13], [14] (2) prioritizing or scoring vulnerabilities using these social media platforms or binary file appearance logs of machines to predict the risk state of machines or systems [7], [11]. These two components have been used in silos and in this paper, we ignore the steps between vulnerability exploit analysis and the final task of real world cyber attack prediction by removing the preconceived notions used in earlier studies where vulnerability exploitation is considered a precursor towards attack prediction. We instead hypothesize on user interaction dynamics conceived through posts surrounding these vulnerabilities in these underground platforms to generate warnings for future attacks. We note that we *do not* consider whether vulnerabilities have been exploited or not in these discussions since a lot of zero-day attacks [11] might occur before such vulnerabilities are even indexed and their gravity might lie hidden in discussions related to other associated vulnerabilities or some discussion on exploits. The premise on which this research is set up is based on the dynamics of all kinds of discussions in the darkweb forums, but we attempt to filter out the noise to mine important patterns by studying whether a piece of information gains traction within important communities.

To this end, the major contributions of this research investigation are as follows:

- We create a network mining technique using the directed reply network of users who participate in the darkweb forums, to extract a set of specialized users we term *experts* whose posts with *popular vulnerability mentions* gain attention from other users in a specific time frame.
- Following this, we generate several time series of features that capture the dynamics of interactions centered around

these *experts* across individual forums as well as general social network and forum posting statistics based feature time series.

- We use these time series features and train a supervised learning model based on logistic regression with attack labels for 2 different events from an organization to predict daily attacks. We obtain the best results with an F1 score of 0.53 on a feature that explores the path structure between *experts* and other users compared to the random (without prior probabilities) F1 score of 0.37. Additionally, we find superior performance of features from discussions that involve vulnerability information over network centralities and forum posting statistics.

The rest of the paper is organized as follows: we introduce several terms and the dataset related to the vulnerabilities and the darkweb in Section II, the general framework for attack prediction including feature curation and learning models in Section III, and finally the experimental evaluations in Section IV.

## II. BACKGROUND AND DATASET

In this section, we describe the dataset used in our research to analyze the interaction patterns of the users in the Darkweb and the real world security incidents<sup>3</sup> data that we use as ground truth for the evaluation of our prediction models.

### A. Enterprise-Relevant External Threats (GT)

We use the Ground Truth (GT) available from the CAUSE program<sup>4</sup> that provided us with data from Armstrong Corporation which contains information on cyber attacks on their systems in the period of April 2016 to September 2017. The data contains the following relevant attributes: { *event-type*: The type of attack called *event-type* and *event occurred date*: Date on which there was an attack of particular event-type. The *event-types* that are used in this study are: *Malicious email* refers to an event associated with an individual in the organization receiving an email that contains either a malicious attachment of link, and *Endpoint Malware* refers to a malware on endpoint that is discovered on an endpoint device. This includes, but not limited to, ransomware, spyware, and adware. As shown in Figure 1, the distribution of attacks over time is different for the events. The total number of incidents reported for the events are as follows: 119 tagged as *endpoint-malware* and 135 for *malicious-email* events resulting in a total of 280 incidents over a span of 17 months that were considered in our study.

### B. Darkweb data

The dark web forms a small part of the deep web, the part of the Web not indexed by web search engines, although sometimes the term deep web is mistakenly used to refer specifically to the dark web. We obtain all darkweb data used in this study through an API provided by a commercial platform<sup>5</sup>.

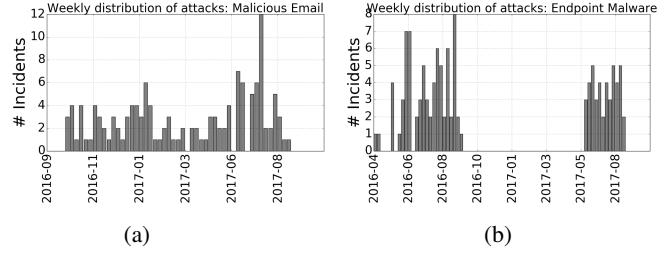


Fig. 1: Weekly occurrence of security breach incidents of different types (a) Malicious email (b) Endpoint Malware

A darkweb forum structure exhibits a hierarchical structure: each forum consists of several independent threads, a thread caters to a particular discussion on a topic, each thread spans several posts initiated by multiple users over time. We note that one user can appear multiple times in the sequence of posts depending on when and how many times the user posted in that thread. However the dataset we obtained does not contain the hierarchical information of reposting - it does not provide us with which user did a particular user, reply to, while posting or replying in a thread. We filter out forums based on a threshold number of posts that were created in the timeframe of January 2016 to September 2017. We gathered data from 179 forums in that time period where the total number of unique posts were 557,689 irrespective of the thread that they belonged to. The number of forums with less than 100 posts is large and therefore we only consider forums which have greater than 5,000 posts in that time period which gave us a total of 53 forums. We denote the set of these 53 forums used in this dataset using the symbol  $F$ .

**Common Vulnerabilities and Exposures (CVE):** The database of Common Vulnerabilities and Exposures maintained on a platform operated by the MITRE corporation<sup>6</sup> provides an identity mapping for publicly known information-security vulnerabilities and exposures. We collect all the information regarding the vulnerability mentions in the darkweb forums in the period from January 2016 to October 2017. The total number of CVEs mentioned in the posts across all forums in this period are 3553.

**CVE - CPE mapping:** A CPE (Common Platform Enumeration) is a structured naming scheme for identifying and grouping clusters of information technology systems, software and packages maintained in a platform NVD (National Vulnerability Database) operated by NIST<sup>7</sup>. Each CVE can be assigned to different CPE groups based on the naming system of CPE families as described in [9]. Similarly, each CPE family can have several CVEs that conform to its vendors and products that the specific CPE caters to. In order to cluster the set of CVEs in our study into a set of CPE groups, we use the set of CPE tags for each CVE from the NVD database maintained by NIST. For the CPE

<sup>3</sup>We would often use the terms attacks/incidents/events interchangeably

<sup>4</sup><https://www.iarpa.gov/index.php/research-programs/cause>

<sup>5</sup>Data is provided by Cyber Reconnaissance, Inc., [www.cyr3con.ai](http://www.cyr3con.ai)

<sup>6</sup><http://cve.mitre.org>

<sup>7</sup><https://nvd.nist.gov/cpe.cfm>

tags, we only consider the operating system platform and the application environment tags for each unique CPE. Examples of CPE would include: *Microsoft Windows\_95*, *Canonical ubuntu\_linux*, *Hp elitebook\_725\_g3*. The first component in each of these CPEs denote the operating system platform and the second component denotes the application environment and their versions.

### III. FRAMEWORK FOR ATTACK PREDICTION

The mechanism for attack predictions can be described in 3 steps : (1) given a time point  $t$  on which we need to predict an enterprise attack of a particular event type (2) we use features from the darkweb forums prior to  $t$  and, (3) we use these features as input to a learned model to predict attack on  $t$ . So one of the main tasks involves learning the attack prediction model, one for each event type. Below we describe steps (2) and (3) - feature curation and building supervised learning models.

#### A. Feature curation

We first describe the mechanism in which we build temporal networks following which we describe the features used for the prediction problem. We build 3 groups of features across forums: (1) Expert centric (2) User/Forum statistics (3) Network centralities.

**Darkweb Reply Network:** We assume the absence of global user IDs across forums<sup>8</sup> and therefore analyze the social interactions using networks induced on specific forums instead of considering the global network across all forums. We denote the directed reply graph of a forum  $f \in F$  by  $G^f = (V^f, E^f)$  where  $V^f$  denotes the set of users who posted or replied in some thread in forum  $f$  at some time in our considered time frame of data and  $E^f$  denotes the set of 3-tuple  $(u_1, u_2, rt)$  directed edges where  $u_1, u_2 \in V^f$  and  $rt$  denotes the time at which  $u_1$  replied to a post of  $u_2$  in some thread in  $f$ ,  $u_1 \rightarrow u_2$  denoting the edge direction. We denote by  $G_\tau^f = (V_\tau^f, E_\tau^f)$ , a temporal subgraph of  $G^f$ ,  $\tau$  being a time window such that  $V_\tau^f$  denotes the set of individuals who posted in  $f$  in that window and  $E_\tau^f$  denotes the set of tuples  $(v_1, v_2, rt)$  such that  $rt \in \tau$ ,  $v_1, v_2 \in V_\tau^f$ . We use 2 operations to create temporal networks: *Create* - that takes a set of forum posts in  $f$  within a time window  $\tau$  as input and creates a temporal subgraph  $G_\tau^f$  and *Merge* - that takes two temporal graphs as input and merges them to form an auxiliary graph. To keep the notations simple, we would drop the symbol  $f$  when we describe the operations for a specific forum in  $F$  as context but which would apply for any forum  $f \in F$ . We describe these two operations in brief, however a detailed algorithm relating the network construction is given in Algorithm 1 of **Appendix A1**<sup>9</sup>. We adopt an incremental analysis approach

<sup>8</sup>Note that even in the presence of global user IDs across forums, a lot of anonymous or malicious users would create multiple profiles across forums and create multiple posts with different profiles, identifying and merging which is an active area of research.

<sup>9</sup>Online Appendix: <http://www.public.asu.edu/~ssarka18/appendix.pdf>

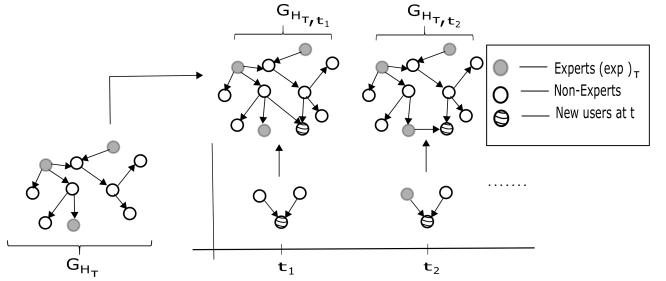


Fig. 2: An illustration to show the *Merge* operation:  $G_{H_\tau}$  denotes the historical network using which the experts shown in gray are computed.  $\{G_{t_1}, G_{t_2}, \dots\}$  denote the networks at time  $t_1, t_2, \dots \in \tau, \tau \in \Gamma$ .

by splitting the entire set of time points in our frame of study into a sequence of time windows  $\Gamma = \{\tau_1, \tau_2, \dots, \tau_Q\}$ , where each subsequence  $\tau_i, i \in [1, Q]$  is equal in time span and non-overlapping and the subsequences are ordered by their starting time points for their respective span.

**CREATE:** *Creating the reply graph* - Let  $h$  be a particular thread or topic within a forum  $f$  containing posts by users  $V_h^f = \{u_1, \dots, u_k\}$  posted at corresponding times  $T_h^f = \{t_1, \dots, t_k\}$ , where  $k$  denotes the number of posts in that thread and  $t_i \geq t_j$  for any  $i > j$ , that is the posts are chronologically ordered. To create the set of edges  $E_h^f$ , we connect 2 users  $(u_i, u_j) \in V_h^f$  such that  $i > j$ , that is user  $u_i$  has potentially replied to  $u_j$ , and subject to a set of *spatial* and *temporal* constraints (**Appendix A1**). These constraints make up for the absence of exact information about the reply hierarchies as to whom  $u$  replied to in a particular post in  $h$ . **MERGE:** *Merging network* - In order to create a time series feature  $\mathcal{T}_{x,f}$  for feature  $x$  from threads in forum  $f$  that maps each time point  $t \in \tau, \tau \in \Gamma$  to a real number, we use 2 networks: (1) the historical network  $G_{H_\tau}$  which spans over time  $H_\tau$  such that  $\forall t' \in H_\tau$ , and  $t \in \tau$ , we have  $t' < t$ , and (2) the network  $G_t^f$  induced by user interactions between users in  $E_t$ , which varies temporally for each  $t \in \tau$ . We note that the historical network  $G_{H_\tau}$  would be different for each subsequence  $\tau$  and same for all  $t \in \tau$ , so as the subsequences  $\tau \in \Gamma$  progress with time, the historical network  $G_{H_\tau}$  also changes, and we discuss the choice of spans  $\tau \in \Gamma$  and  $H_\tau$  in Section IV. Finally, for computing feature values for each time point  $t \in \tau$ , we merge the 2 networks  $G_{H_\tau}$  and  $G_t$  to form the auxiliary network  $G_{H_\tau, t} = (V_{H_\tau, t}, E_{H_\tau, t})$ , where  $V_{H_\tau, t} = V_{H_\tau} \cup V_t$  and  $E_{H_\tau, t} = E_{H_\tau} \cup E_t$ . A visual illustration of this method is shown in Figure 2. Now we describe the several features we used that would be fed to a learning model for attack prediction. We compute time series of several features  $x, \mathcal{T}_{x,f}[t]$  for every time point  $t$  in our frame of study and for every forum  $f$  separately.

#### 1. Expert centric features

We extract a set of users we term *experts* who have a history of CVE mentions in their posts and whose posts have

Group	Features	Description
Expert centric	Graph Conductance	$\tau_x[t] = \frac{\sum_{x \in exp_\tau} \sum_{y \in V_t \setminus exp_\tau} \pi(exp_\tau) P_{xy}}{\pi(exp_\tau)}$ where $\pi(\cdot)$ is the stationary distribution of the network $G_{H_\tau, t}$ , $P_{xy}$ denotes the probability of random walk from vertices $x$ to $y$ . The conductance represents the probability of taking a random walk from any of the <i>experts</i> to one of the users in $V_t \setminus exp_\tau$ , normalized by the probability weight of being on an expert.
	Shortest Path	$\tau_x[t] = \frac{1}{ exp_\tau } \sum_{e \in exp_\tau} \min_{u \in V_t \setminus exp_\tau} s_{e,u}$ where $s_{e,u}$ denotes the shortest path from an expert $e$ to user $u$ following the direction of edges.
	Expert replies	$\tau_x[t] = \frac{1}{ exp_\tau } \sum_{e \in exp_\tau}  OutNeighbors(e) $ where $OutNeighbors(\cdot)$ denotes the out neighbors of user in the network $G_{H_\tau, t}$ .
	Common Communities	$\tau_x[t] = \{N(c(u) \mid c(u) \in c_{experts} \wedge u \in V_t \setminus exp_\tau)\}$ where $c(u)$ denotes the community index of user $u$ , $c_{experts}$ that of the experts and $N(\cdot)$ denotes a counting function. It counts the number of users who share communities with experts.
Forum/User Statistics	Number of threads	$\tau_x[t] =  \{h \mid \text{thread } h \text{ was posted on } t\} $
	Number of users	$\tau_x[t] =  \{u \mid \text{user } u \text{ posted on } t\} $
	Number of expert threads	$\tau_x[t] =  \{h \mid \text{thread } h \text{ was posted on } t \text{ by users } u \in \text{experts}\} $
	Number of CVE mentions	$\tau_x[t] =  \{CVE \mid \text{CVE was mentioned in some post on } t\} $
Network Centralities	Outdegree <sub>k</sub>	$\tau_x[t] = \text{Average value of top } k \text{ users, by outdegree on } t$
	Outdegree <sub>k</sub> CVE	$\tau_x[t] = \text{Average value of top } k \text{ users with more than 1 CVE mention in their posts, by outdegree on } t$
	Pagerank <sub>k</sub>	$\tau_x[t] = \text{Average value of top } k \text{ users, by Pagerank on } t$
	Pagerank <sub>k</sub> CVE	$\tau_x[t] = \text{Average value of top } k \text{ users with more than 1 CVE mention in their posts, by pagerank on } t$
	Betweenness <sub>k</sub>	$\tau_x[t] = \text{Average value of top } k \text{ users, by Betweenness on } t$
	Betweenness <sub>k</sub> CVE	$\tau_x[t] = \text{Average value of top } k \text{ users with more than 1 CVE mention in their posts, by betweenness on } t$

TABLE I: List of features used for learning. Each feature  $\tau_x$  is computed separately across forums.

gained attention in terms of replies. Following that, we mine several features that explain how attention is broadcast by these *experts* to other posts. All these features are computed using the auxiliary networks  $G_{H_\tau, t}$  for each time  $t$ . Our hypothesis is based on the premise that any unusual activity must spur attention from users who have knowledge about vulnerabilities.

We focus on users whose posts in a forum contain most discussed CVEs belonging to *important CPEs* over the timeframe of analysis, where the importance will shortly be formalized. For each forum  $f$ , we use the historical network  $G_{H_\tau}^f$  to extract the set of *experts* relevant to timeframe  $\tau$ , that is  $exp_\tau^f \in V_{H_\tau}^f$ . First, we extract the top CPE groups  $CP_\tau^{top}$  in the time frame  $H_\tau$  based on the number of historical mentions of CVEs. We sort the CPE groups based on the sum of the CVE mentions in  $\tau$  that belong to the respective CPE groups and take the top 5 CPE groups by sum in each  $H_\tau$ . Using these notations, the experts  $exp_\tau^f$  from history  $H_\tau$  considered for time span  $\tau$  are defined as users in  $f$  with the following three constraints: (1) Users who have mentioned a CVE in their post in  $H_\tau$ . This ensures that the user engages in the forums with content that is relevant to vulnerabilities. (2) let  $\theta(u)$  denote the set of CPE tags of the CVEs mentioned by user  $u$  in his/her posts in  $H_\tau$  and such that it follows the constraint: either  $\theta(u) \in CP_\tau^{top}$  where the user's CVEs are grouped in less than 5 CPEs or,  $CP_\tau^{top} \in \theta(u)$  in cases where a user has posts with CVEs in the span  $H_\tau$ , grouped in more than 5 CPEs. This constraint filters out users who discuss vulnerabilities which are not among the top CPE groups in  $H_\tau$  and (3) the in-degree of the user  $u$  in  $G_{H_\tau}$  should cross a threshold. This constraint ensures that there are a significant number of users who potentially responded to this user thus establishing  $u$ 's central position in

the reply network. Essentially, these set of experts  $exp_\tau$  from  $H_\tau$  would be used for all the time points in  $\tau$ . We curate path and community based features based on these experts listed in Table I. These expert-centric features try to quantify the distance between an expert and a daily user(non-expert) in terms of how fast a post from that user receives attention from the expert. In that sense, the community features also measure the like-mindedness of non-experts and experts.

**Why focus on experts?** To show the significance of these properties in comparison to other users, we perform the following hypothesis test: we collect the time periods of 3 widely known security events: the WannaCry ransomware attack that happened on May 12, 2017 and the vulnerability MS-17-010, the Petya cyber attack on 27 June, 2017 with the associated vulnerabilities CVE-2017-0144, CVE-2017-0145 and MS-17-010, the Equifax breach attack primarily on March 9, 2017 with vulnerability CVE-2017-5638. We consider two sets of users across all forums -  $exp_\tau$ , where  $G_{H_\tau}$  denotes the corresponding historical network prior to  $\tau$  in which these 3 events occurred and the second set of users being all  $U_{alt}$  who are not experts and who fail either one of the two constraints: they have mentioned CVEs in their posts which do not belong to  $CP_\tau^{top}$  or their in-degree in  $G_{H_\tau}$  lies below the threshold. We consider  $G_{H_\tau}$  being induced by users in the last 3 weeks prior to the occurrence week of each event for both the cases, and we consider the total number of interactions ignoring the direction of reply of these users with other users. Let  $\deg_{exp}$  denote the vector of count of interactions in which the *experts* were involved and  $\deg_{alt}$  denote the vector of counts of interactions in which the users in  $U_{alt}$  were involved. We randomly pick number of users from  $U_{alt}$  equal to the number of experts and sort the vectors by count. We conduct

a 2 sample t-test on the vectors  $\text{deg}_{\text{exp}}$  and  $\text{deg}_{\text{alt}}$ . The null hypothesis  $H_0$  and the alternate hypothesis  $H_1$  are defined as follows;  $H_0 : \text{deg}_{\text{exp}} \leq \text{deg}_{\text{alt}}$ ,  $H_1 : \text{deg}_{\text{exp}} > \text{deg}_{\text{alt}}$ . The null hypothesis is rejected at significance level  $\alpha = 0.01$  with  $p$ -value of 0.0007. This suggests that with high probability, experts tend to interact more prior to important real world cybersecurity breaches than other users who randomly post CVEs.

Now, we conduct a second  $t$ -test where we randomly pick 4 weeks not in the weeks considered for the data breaches, to pick users  $U_{\text{alt}}$  with the same constraints. We use the same hypotheses as above and when we perform statistical tests for significance, we find that the null hypothesis is not rejected at  $\alpha=0.01$  with a  $p$ -value close to 0.05. This empirical evidence from the  $t$ -test also suggests that the interactions with  $\text{exp}_\tau$  are more correlated with an important cybersecurity incident than the other users who post CVEs not in top CPE groups and therefore it is better to focus on users exhibiting our desired properties as experts for cyber attack prediction. Note that the  $t - test$  evidence also incorporates a special temporal association since we collected events from three interleaved timeframes corresponding to the event dates.

## 2. User/Forum Statistics Features

We try to see whether the forum or user posting statistics are themselves any indicators of future cyber attacks - for this we compute *Forum/User Statistics* as described in Table I.

## 3. Network centralities Features

In addition, we also tested several network *Centrality* features mentioned in Table I. The purpose is to check whether emergence of central users in the reply network  $G_t$ ,  $t \in \tau$ , are good predictors of cyber attacks. We note that in this case, we only use the daily reply networks to compute the features unlike the expert centric network features where we use  $G_{H_\tau, t}$ .

## B. Learning Models for Prediction

In this section we explain how we use the time series data  $\mathcal{T}_{x,f}$  to predict an attack at any given time point  $t$ . We consider a supervised learning model in which the time series  $\mathcal{T}_x$  is formed by averaging  $\mathcal{T}_{x,f}$  across all forums in  $f \in F$  at each time point  $t$  and then using them for the prediction task. We treat the attack prediction problem in this paper as a binary classification problem in which the objective is to predict whether there would be an attack at a given time point  $t$ . Since the incident data in this paper contains the number of incidents that occurred at time point  $t$ , we assign a label of 1 for  $t$  if there was at least one attack at  $t$  and 0 otherwise.

In [4], the authors studied the effect of longitudinal sparsity in high dimensional time series data, where they propose an approach to assign weights to the same features at different time spans to capture the temporal redundancy. We use 2 parameters:  $\delta$  that denotes the start time prior to  $t$  from where we consider the features for prediction and  $\eta$ , the time span for the features to be considered. An illustration is shown in Figure 3 where to predict an attack occurrence at time  $t$ , we use the features for each time  $t_h \in [t_{-\eta-\delta}, t_{-\delta}]$ . Here we use

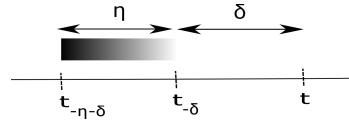


Fig. 3: Temporal feature selection window for predicting an attack at time  $t$

logistic regression with longitudinal ridge sparsity that models the probability of an attack as follows with  $\mathbf{X}$  being the set of features and  $\beta$  being the vector of coefficients:

$$P(\text{attack}(t) = 1 | \mathbf{X}) = \frac{1}{1 + e^{-(\beta_0 + \sum_{k=\eta+\delta}^{\delta} \beta_k x_{t-k})}} \quad (1)$$

The final objective function to minimize over  $N$  instances where  $N$  here is the number of time points spanning the attack time frame is :  $l(\beta) = -\sum_{i=1}^N (y_i(\beta_0 + \mathbf{x}_i^T \beta) - \log(1 + \exp^{\beta_0 + \mathbf{x}_i^T \beta}) + \lambda \beta^T \beta)$ ,  $y$  being the instance label.

One of the major problems of the dataset is the imbalance in the training and test dataset as will be described in Section IV, so in order to use all features in each group together for prediction, we use 3 additional regularization terms: the L1 penalty, the L2 penalty and the *Group Lasso* regularization [5]. The final objective function can be written as:

$$l(\beta) = -\sum_{i=1}^N \log(1 + e^{-y_i(\beta^T \mathbf{x}_i)}) + \frac{m}{2} \|\beta\|_2^2 + l \|\beta\|_1 + g.GL(\beta) \quad (2)$$

where  $m$ ,  $l$  and  $g$  are the hyper-parameters for the regularization terms and the  $GL(\beta)$  term is  $\sum_{g=1}^G \|\beta_{\mathcal{I}_g}\|_2$ , where  $\mathcal{I}_g$  is the index set belonging to the  $g^{\text{th}}$  group of variables,  $g = 1 \dots G$ . Here each  $g$  is the time index  $t_h \in [t_{-\eta-\delta}, t_{-\delta}]$ , so this group variable selection selects all features of one time in history while reducing some other time points to 0. It has the attractive property that it does variable selection at the temporal group level and is invariant under (group-wise) orthogonal transformations like ridge regression. We note that while there are several other models that could be used for prediction that incorporates the temporal and sequential nature of the data like hidden markov models (HMM) and recurrent neural networks (RNN), the logit model allows us to transparently adjust to the sparsity of data, specially in the absence of a large dataset.

## IV. EXPERIMENTAL EVALUATIONS

In our work, the granularity for each time index in the  $\mathcal{T}$  function is 1 day, that is we compute feature values over all days in the time frame of our study. For incrementally computing the values of the time series, we consider the time span of each subsequence  $\tau \in \Gamma$  as 1 month, and for each  $\tau$ , we consider  $H_\tau = 3$  months immediately preceding  $\tau$ . That is, for every additional month of training or test data that is provided to the model, we use the preceding 3 months to create the historical network and compute the corresponding features on all days in  $\tau$ . For choosing the experts with an in-degree threshold, we select a threshold of 10 to filter out users having in-degree less than 10 in  $G_{H_\tau}$  from  $\text{exp}_\tau$ . For the centralities

features, we set  $k$  to be 50, that is we choose the top 50 users sorted by that corresponding metric in Table I. We build different learning models using the ground truth available from separate *event – types*.

As mentioned in Section III-B, we consider a binary prediction problem in this paper - we assign an attack flag of 1 for at least 1 attack on each day and 0 otherwise have the following statistics: for *malicious-email*, out of 335 days considered in the dataset, there have been reported attacks on 97 days which constitutes a positive class ratio of around 29%, for *endpoint-malware* the total number of attack days are 31 out of 306 days of considered span in the training dataset which constitutes a positive class ratio of around 26%. For evaluating the performance of the models on the dataset, we split the time frame of each event into 70%-30% averaged to the nearest month separately for each *event – type*. That is we take the first 80% of time in months as the training dataset and the rest 20% in sequence for the test dataset. We avoid shuffle split as generally being done in cross-validation techniques in order to consider the consistency in using sequential information when computing the features. As shown in Figures 1, since the period of attack information provided varies in time for each of the events, we use different time frames for the training model and the test sets. For the event *malicious email* which remains our primary testbed evaluation event, we consider the time period from October 2016 to June 2017 (9 months) in the Darkweb forums for our training data and the period from July 2017 to August 2017 (3 months) as our test dataset, for the *endpoint – malware*, we use the time period from April 2016 to September 2016 (6 months) as our training time period and June 2017 to August 2017 (3 months) as our test data for evaluation.

We consider a span of 1 week time window  $\eta$  while keeping  $\delta = 7$  days. From among the set of statistics features that were used for predicting *malicious – email* attacks shown in Figure 4(e), we observe the best results using the number of threads as the signal for which we observe a precision of 0.43, recall of 0.59 and an F1 score of 0.5 against the random F1 of 0.34 for this type of attacks. From among the set of expert-centric features in Figure 4(a), we obtain the best results from graph conductance with a precision of 0.44, recall of 0.65 and an F1 score of 0.53 which shows an increase in recall over the number of threads measure. Additionally, we observe that the best features in terms of F1 score are graph conductance and shortest paths whereas number of threads and vulnerability mentions turn out to be the best among the statistics. For the attacks belonging to the type *endpoint – malware*, we observe similar characteristics for the expert-centric features in Figure 4(b) where we obtain a best precision of 0.34, recall of 0.74 and an F1 score of 0.47 against a random F1 of 0.35, followed by the shortest paths measure. However for the statistics measures we obtain a precision of 0.35, recall 0.61 and an F1 score of 0.45 for the vulnerability mentions followed by the number of threads which gives us an F1 score of 0.43. Although the common communities features doesn't help much in the overall prediction results, in the

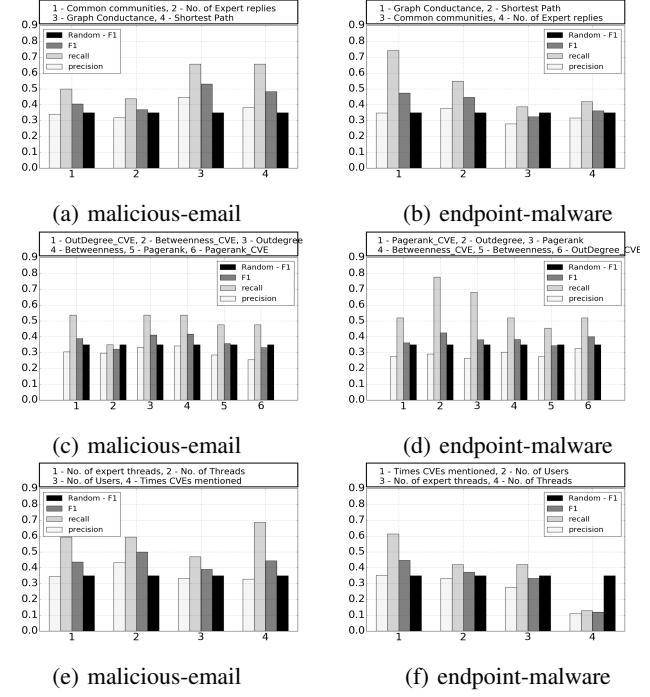


Fig. 4: Classification results for the features considering the logistic regression model:  $\delta = 7$  days,  $\eta = 8$  days.

following section we describe a special case that demonstrates the predictive power of the community structure in networks. On the other hand, when we investigate the centralities features with respect to the prediction performance in Figure 4(c), we find that just looking at network centralities does not help. The best values we obtain for *malicious-email* event predictions are from the outdegree and betweenness metrics both of which gives us an F1 score of 0.41. Surprisingly, we find that when the metrics are used for only the users with CVE mentions, the results are worse with the best F1 score for outdegree CVE having an F1 score of 0.38. This calls for more complex understanding of path structures between users than just focusing on user significance solely. The challenging nature of the supervised prediction problem is not just due to the issue of class imbalance, but also the lack of large samples in the dataset which if present, could have been used for sampling purposes. As an experiment, we also used Random Forests as the classification model, but we did not observe any significant improvements in the results over the random case.

For the model with the Group lasso regularization in Equation 2, we set the parameters  $m, l, g$  and 0.3, 0.3 and 0.1 respectively. We obtain better results for each group of features together on the *malicious-email* events with an F1 score of 0.55 for Expert centric, 0.51 with Forum/user statistics and 0.49 with network centrality based features.

#### *Prediction in High Activity Weeks*

One of the main challenges in predicting external threats without any method to correlate them with external data sources like darkweb or any other database is that it is difficult

to validate which kinds of attacks are most correlated with these data sources. To this end, we examine a controlled experiment setup for the *malicious – email* attacks in which we only consider the weeks which exhibited high frequency of attacks compared to the overall timeframe: in our case we consider weeks having more than 5 attacks in test time frame. These high numbers may be due to multiple attacks in one or few specific days or few attacks on all days. We run the same supervised prediction method but evaluate them only on these specific weeks.

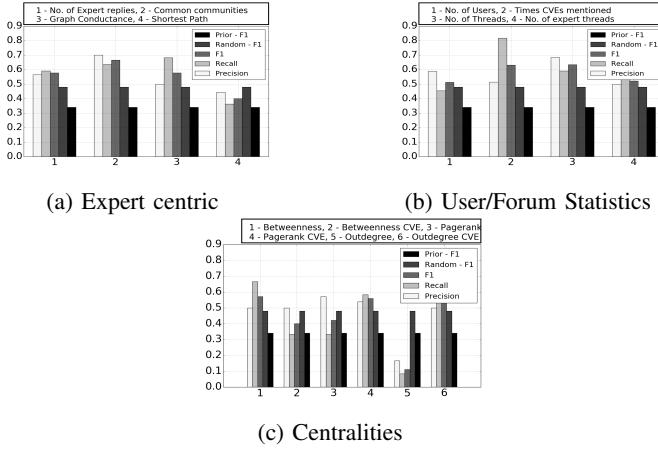


Fig. 5: Classification results for *malicious – email* attacks in high frequency weeks,  $\delta = 7$  days and  $\eta = 8$  days.

From the results shown in Figure 5, we find that the best results were shown by the common communities feature having a precision of 0.7 and a recall of 0.63 and an F1 score of 0.67 compared to the random (no priors) F1 score of 0.48 and a random (with priors) F1 score of 0.34 for the same time parameters. Among the statistics measures, we obtained a highest F1 score of 0.63 for the vulnerability mentions feature. From among the set of centralities features, we find that betweenness measure has the best F1 score of 0.58 with a precision of 0.5 and a recall of 0.78. This also suggests the fact that analyzing the path structure between nodes is useful since betweenness relies on the paths passing through a node. Additionally, we find unlike the results over all the days, for these specific weeks, the model achieves high precision while maintaining comparable recall emphasizing the fact that the number of false positives are also reduced during these periods. This correlation between the weeks that exhibit huge attacks and the prediction results imply that the network structure analytics can definitely help generate alerts for cyber attacks.

## V. RELATED WORK AND CONCLUSION

Using network analysis to understand the topology of Darkweb forums has been studied at breadth in [6] where the authors use social network analysis techniques on the reply networks of forums. There have been several attempts to use external social media data sources to predict real world cyber

attacks [2], [7], [8]. Using machine learning models to predict security threats [2] has many open research fields including predicting whether a vulnerability would be exploited based on Darkweb sources [3], [9]. The availability of large external data sources makes the case for using machine learning methods for cyber attack prediction more promising. Previous studies also include using time series models for forecasting the number of cyber incidents [16] which motivates the need of such models for cyber attack prediction. The authors in [17] look at text mining techniques to understand the content of the posts in various social media platforms that provide threat intelligence. In this study, we argue that the darkweb can be a reliable source of information for predicting external enterprise threats. We leverage the network and interaction patterns in the forums to understand the extent to which they can be used as useful indicators. Our study also opens further research possibilities surrounding sentiment analysis on these discussions that could help track the malicious discussions and hence defend against cyber conflict during competition.

## REFERENCES

- [1] Samtani, Sagar, Ryan Chinn, and Hsinchun Chen. "Exploring hacker assets in underground forums." IEEE (ISI), 2015.
- [2] Liu, Yang, et al. "Cloudy with a Chance of Breach: Forecasting Cyber Security Incidents." USENIX Security Symposium. 2015.
- [3] Nunes, Eric, et al. "Darknet and deepnet mining for proactive cybersecurity threat intelligence." IEEE ISI (2016).
- [4] Xu, Tingyang, Jiangwen Sun, and Jinbo Bi. "Longitudinal lasso: Jointly learning features and temporal contingency for outcome prediction." ACM,KDD 2015.
- [5] Meier, Lukas, Sara Van De Geer, and Peter Bühlmann. "The group lasso for logistic regression." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70.1 (2008): 53-71.
- [6] Almukaynizi, Mohammed, et al. "Predicting cyber threats through the dynamics of user connectivity in darkweb and deepweb forums." ACM Computational Social Science. (2017).
- [7] Liu, Yang, et al. "Predicting cyber security incidents using feature-based characterization of network-level malicious activities." 2015 ACM International Workshop Security and Privacy Analytics.
- [8] Khandpur, Rupinder Paul, et al. "Crowdsourcing cybersecurity: Cyber attack detection using social media." ACM CIKM 2017.
- [9] Almukaynizi, Mohammed, et al. "Proactive identification of exploits in the wild through vulnerability mentions online." IEEE CyCON, 2017.
- [10] Thonnard, Olivier, et al. "Are you at risk? Profiling organizations and individuals subject to targeted attacks." International Conference on Financial Cryptography and Data Security. Springer 2015.
- [11] Bilge, Leyla, and Tudor Dumitras. "Before we knew it: an empirical study of zero-day attacks in the real world." Proceedings of the 2012 ACM conference on Computer and communications security.
- [12] Sabottke, Carl, Octavian Suciu, and Tudor Dumitras. "Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits." USENIX Security Symposium. 2015.
- [13] Herley, Cormac, and Dinei Florêncio. "Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy." Economics of information security and privacy. Springer, Boston, MA, 2010. 33-53.
- [14] Allodi, Luca, Marco Corradin, and Fabio Massacci. "Then and now: On the maturity of the cybercrime markets the lesson that black-hat marketeers learned." IEEE Transactions on Emerging Topics in Computing 4.1 (2016): 35-46.
- [15] Chen, Hsinchun. "Sentiment and affect analysis of dark web forums: Measuring radicalization on the internet." IEEE ISI 2008.
- [16] Okutan, Ahmet, et al. "POSTER: Cyber Attack Prediction of Threats from Unconventional Resources (CAPTURE)." Proceedings of the 2017 ACM SIGSAC.
- [17] Sapienza, Anna, et al. "Early warnings of cyber threats in online discussions." Data Mining Workshops (ICDMW), 2017.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/366527881>

# End-to-end AI Framework for Hyperparameter Optimization, Model Training, and Interpretable Inference for Molecules and Crystals

Preprint · December 2022

DOI: 10.48550/arXiv.2212.11317

---

CITATIONS

0

READS

6

6 authors, including:



Ruijie Zhu

Northwestern University

3 PUBLICATIONS 0 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Molecular Dynamics [View project](#)

# End-to-end AI Framework for Interpretable Prediction of Molecular and Crystal Properties

Hyun Park<sup>1,3,7</sup>, Ruijie Zhu<sup>2,3</sup>, E. A. Huerta<sup>3,4,5</sup>, Santanu Chaudhuri<sup>3,6</sup>, Emad Tajkhorshid<sup>1,7,8</sup> and Donny Cooper<sup>9</sup>

<sup>1</sup> Theoretical and Computational Biophysics Group, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

<sup>2</sup> Department of Materials Science and Engineering, Northwestern University, Evanston, Illinois 60208, USA

<sup>3</sup> Data Science and Learning Division, Argonne National Laboratory, Lemont, Illinois 60439, USA

<sup>4</sup> Department of Computer Science, University of Chicago, Chicago, Illinois 60637, USA

<sup>5</sup> Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

<sup>6</sup> Multiscale Materials and Manufacturing Lab, University of Illinois Chicago, Chicago, Illinois 60607, USA

<sup>7</sup> Department of Biochemistry, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

<sup>8</sup> Center for Biophysics and Quantitative Biology, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

<sup>9</sup> Computational Science and Engineering, Data Science and AI Department, TotalEnergies EP Research & Technology USA, LLC, Houston, Texas 77002 USA

E-mail: hyunp2@illinois.edu

December 2022

**Abstract.** We introduce an end-to-end computational framework that allows for hyperparameter optimization using the DeepHyper library, accelerated model training, and for interpretable AI inference. The framework is based on state-of-the-art AI models including CGCNN, PhysNet, SchNet, MPNN, MPNN-transformer, and TorchMD-NET. We employ these AI models along with the benchmark QM9, hMOF, and MD17 datasets to showcase how the model can predict user-specified material properties within modern computing environments. We demonstrate translational applications in the modeling of small molecules, inorganic crystals and nanoporous metal organic frameworks with a unified, standalone framework. We have deployed and tested this framework in the ThetaGPU supercomputer at the Argonne Leadership Computing Facility (ALCF), and in the Delta supercomputer at the National Center for Supercomputing Applications (NCSA) to provide researchers with modern tools to conduct accelerated AI-driven discovery in leadership-class computing environments.

## 1. Introduction

With the explosion of AI models [1, 2, 3, 4, 5] developed to predict various material properties over the recent years, it has become difficult to keep track of the available AI models and the datasets that are used for training and inference. Numerous efforts [6, 7] have been made toward the integration of AI models and their associated datasets in one place to streamline their use for a wide range of applications and a broad community of users [8, 9, 10]. AI models and datasets are often available through open repositories, in the best scenario, so a user can download, deploy and reproduce their putative capabilities. Unfortunately, this is a time-consuming and laborious process, which can be further complicated when tools and libraries used to develop the AI models are not available, deprecated, or non-backwards compatible in computing environments of new users. Furthermore, most of the existing packages are specialized in predicting quantum mechanical (QM) properties of small molecules, few of them support crystals.

In order to address these shortcomings, here we report the construction of a computational framework that consolidates libraries, AI models and AI interpretability tools to study molecules, crystals, and metal-organic frameworks. The framework enables hyperparameter tuning through the open source library `DeepHyper` [11], model training, and interpretable inference of small-molecule QM properties from public datasets such as `QM9` [12], and crystal properties from datasets such as `hMOF` [13].

Key aspects of this computational framework include:

**Novel features of AI models.** The node and edge embedding schemes of two graph neural network models, `PhysNet` and `CGCNN`, were modified from the original adjacency matrix format to an adjacency list format to reduce redundant information and enable faster training. We also adapted small-molecule property prediction models to take in crystal structures as input such as the crystal version of `SchNet`.

**Translational AI applications.** We demonstrate the transferability of the learned force fields by training `TorchMD-NET` model using selected molecular dynamics (MD) trajectory data of a given set of molecules in the `MD17` dataset [14] to perform MD simulations of similar molecules. In particular, we show that a model trained based on ethanol is transferable to both *n*-propanol and iso-propanol, and a model trained using uracil is transferable to pyrimidine and naphthalene. All of the results are automatically logged to weights and biases (`WandB`) [15], a machine learning tracking tool, for simple access.

**Interpretable AI inference.** To gain a better understanding of the model predictions, we provide two novel functionalities to explain the learned features. First, by mapping the last hidden layer of the model onto a 3D plane via the UMAP method, we can make more sense of the molecular clusters with similar properties. Second, by highlighting selected atoms of molecules via the Grad-CAM method, we can now identify which atoms are significant for model predictions.

We expect that this collection of state-of-the-art graph neural networks, transformer models, and analysis methods for small molecules, crystals and metal-organic-frameworks will empower AI practitioners to seamlessly perform hyperparameter optimization, accelerated training and inference, and interpretable AI in modern computing environments with a unified, standalone computational framework.

## 2. Related work

Graph neural networks have shown great success for modeling molecular and crystal structures. For small molecules, a suite of models have been proposed, including DimeNet [4], GemNet [16], SchNet [1] and PhysNet [2]. These models take in atomic coordinates and atomic numbers as input, and represent atoms as nodes and bonds as edges. Typical target properties for these models are QM molecular properties such as internal energy, heat capacity and zero point vibrational energy. For crystal structures, periodic boundary conditions need to be considered, therefore crystal graph representations are typically used. Example graph neural networks that take in crystal structures as input include ALIGNN [17], CGCNN [5], and MEGNet [18]. These models first extract crystal graphs from the structures, then generate atomic and edge embeddings for the center atoms and their neighbors. The bond and edge information is then updated via message passing. The target properties for these models are similarly QM properties of crystals, e.g., formation energy and band gap.

The growing number of the graph neural networks available for this purpose pushes the need for an end-to-end AI framework. Previous efforts toward such a goal typically missed one or more important aspects. For example, MatDeepLearn [7] integrates a suite of graph neural networks, including CGCNN, MEGNet, MPNN, GCN and SchNet. Although it can be used for hyperparameter tuning, model training, and inference, it lacks the explainability feature, which limits the amount of chemical insights that could be extracted from the results. Another example is Dive in Graphs (DIG) [6], which enables model training and explanation. However, it does not allow for hyperparameter tuning, therefore only models with preset hyperparameters can be used. A complete package offering all of the aforementioned functionalities is therefore needed.

Our AI framework also offers the functionality to perform molecular dynamics for small molecules, enabled by TorchMD-NET [3], an SE3-equivariant transformer interatomic potential model that establishes a relationship between atomic configurations and potential energies and forces. The MD trajectories of selected molecules taken from the MD17 dataset were used for training the TorchMD-NET models.

## 3. Methods

Here we describe the key building blocks of our general-purpose AI framework:

- (i) It provides built-in datasets and neural networks that we modified to take in adjacency list format node and edge embeddings, a more efficient embedding scheme

than adjacency matrix format

- (ii) It enables distributed hyperparameter tuning of neural networks via the scalable and computationally efficient library **DeepHyper**
- (iii) Model training and interpretable inference are performed by specifying a few command line arguments
- (iv) Results are auto-logged to **WandB**, a machine learning tool for easy tracking and visualization
- (v) MD simulations can be performed for small molecules using **TorchMD-NET** if trained with MD trajectories from the **MD17** dataset, enabled by the atomic simulation environment (**ASE**) library [19].

This framework has been deployed and tested in leadership computing platforms to reduce the overhead for researchers that require access to hyperparameter tuning, model training and explainable inference tools in a single, unified framework. Below we describe each of these components in further detail.

*Hyperparameter tuning.* This feature was done using the **DeepHyper** [11] library. In this method, hyperparameters of interest are given prior distributions and their posterior distributions are adjusted based on the Centralized Bayesian Optimization (CBO) algorithm with a given acquisition function and a surrogate model. The graph neural networks in this framework are coupled with **DeepHyper** to enable faster hyperparameter tuning.

*Datasets* **QM9** and **MD17** datasets were used as input to graph neural networks. The **QM9** dataset consists of molecular structures and QM properties of 133,885 molecules with up to nine atoms of type H, C, O, N and F. For demonstration purposes, the selected QM properties in this work include the highest occupied molecular orbital (**HOMO**), and zero point vibrational energy (**ZPVE**). The **MD17** dataset consists of ab-initio MD trajectories of 10 molecules at different levels of theory. Both datasets are available in the **PyTorch Geometric** library.

*Node and edge embedding schemes.* Instead of using the original adjacency matrix format for node and edge embeddings, we modified them to adjacency list format. The term embedding, for both molecular and crystal graphs, refers to the information attached to a node (an atom) or an edge (a bond). Both node and edge embeddings can be scalars, vectors or higher order tensors. The node embeddings encode information such as mass, charge and orbital hybridization, whereas the edge embeddings encode information such as interatomic distance and bond order. Depending on the model architecture, some embeddings are physics or chemistry based while others are learned. For physics- or chemistry-based embeddings, the information such as hybridization, mass, atomic radius, and whether the fragment is a part of an aromatic ring is encoded. On the other hand, learned embeddings refer to the embeddings that are optimized by a neural network model via stochastic gradient descent.

In adjacency matrix format edge embeddings, the adjacency matrix is encoded into a fixed-size matrix, whose size is determined by the largest molecule in the dataset. For other molecules, their vectors are padded to be the same dimension as the largest one. Each element in the adjacency matrix indicates whether the two corresponding nodes are connected, as determined via some distance-based criteria. Since padding is applied to smaller molecules in the matrix, users need to know *a priori* the largest molecule size, then perform masking to obtain the padding values, which can be burdensome for GPU memories. By using the adjacency list format, however, only the information for connected atoms is preserved, thereby avoiding the need for padding and taking less memory to load. In this case, faster model training and inference loss convergence speed and higher accuracy are expected. The adjacency list format has been implemented in a number of `Python` libraries such as `Deep Graph Library` (`DGL`) [20] and `PyTorch Geometric` [21]. We will use the `CGCNN` model as an example to demonstrate a boost in model training performance when an adjacency list format is used in place of an adjacency matrix format.

Our AI framework allows users to perform hyperparameter tuning, model training and interpretable inference for pre-trained models or train new models with a few arguments passed. The main improvements over previous general-purpose machine learning model training libraries is the explainability feature, which consists of two parts. First, by extracting high dimensional hidden layer information from the learned models and projecting it onto low dimensions via the uniform manifold approximation and projection (`UMAP`) technique, we can effectively visualize the clustering of molecules, with similar practice as in [22, 23, 24]. Second, Saliency, CAM and Grad-CAM methods are used to highlight important atoms in molecular graphs, as described in [25].

## 4. Results

Below we present a comprehensive analysis of our results, from hyperparameter optimization to interpretable AI inference.

### 4.1. Hyperparameter Optimization

The `DeepHyper` library was used for hyperparameter tuning of graph neural networks. `DeepHyper` is easy to use and can be readily deployed on GPU-based high-performance computing platforms. CPUs can be used if GPUs are not available. However, if the user has access to multiple GPUs, then the GPU option will be automatically chosen, with each core performing hyperparameter search using the CBO algorithm, given an acquisition function such as the upper confidence bound, and a surrogate model, e.g., random forest.

The list of hyperparameters considered in this work along with their ranges are summarized in Table 1. Hyperparameter tuning results for `PhysNet` with ZPVE as target property are shown in Tables 2 and 3. It is worth mentioning that since `DeepHyper` tries

to maximize the objective of search, the opposite number of validation error was used as the objective, therefore a larger absolute value of objective corresponds to a better combination of hyperparameters. The hyperparameter tuning results for PhysNet with HOMO as the target property are shown in Tables A1 and A2

**Table 1.** List of hyperparameters and their ranges.

hyperparameter	log scale	range
<code>agb</code>	true	[1,20]
<code>amp</code>	false	[true,false]
<code>batch_size</code>	/	[128,512]
<code>epochs</code>	true	[10,100]
<code>gradient_clip</code>	true	[1e-05,2]
<code>learning_rate</code>	true	[1e-3,1]
<code>optimizer</code>	/	[SGD,TorchAdam,Adam,LAMB]
<code>weight_decay</code>	true	[2e-6,0.02]

Among the hyperparameters, `agb`—accumulated grad batches—helps overcome memory constraints; `amp`—automatic mixed precision— speeds up neural network training; and `gradient clip`, a machine learning technique where the gradient of neural network parameters is re-scaled by a coefficient between 0 and 1, is known to stabilize neural network training by avoiding sudden changes in parameter values (also known as exploding gradient problem) [26].

**Table 2.** Top 10 DeepHyper hyperparameter combinations for PhysNet with ZPVE as target property.

<code>agb</code>	<code>amp</code>	<code>batch_size</code>	<code>gradient_clip</code>
4	TRUE	190	0.00245
3	TRUE	190	0.00022
1	TRUE	397	0.00032
4	TRUE	196	0.00122
3	TRUE	174	1.60E-05
4	TRUE	154	3.25E-05
4	TRUE	300	0.732
4	TRUE	228	0.00389
2	FALSE	359	0.72537
11	TRUE	168	0.00243

The optimal hyperparameter combinations found by DeepHyper are listed in the top rows of Tables 2 and 3. The optimal objective is -0.9226. We notice that the optimal hyperparameters include f32 precision (`amp="false"`), a standard learning rate (0.00296), and a low gradient norm clipping value (0.00245). These result in small gradient accumulation, which may help mitigate sudden gradient updates.

**Table 3.** As Table 2 for the rest of parameters optimized through DeepHyper.

learning_rate	optimizer	weight_decay	objective
0.00296	torch_adam	1.03E-05	-0.9226
0.75169	torch_adam	5.14E-06	-6.7526
0.00015	lamb	2.69E-06	-6.7925
0.32274	lamb	1.45E-05	-7.1168
0.17673	lamb	7.80E-06	-12.31
0.00144	torch_adam	1.13E-05	-15.394
0.02986	lamb	3.57E-06	-26.13
0.00966	torch_adam	7.16E-06	-27.627
0.02491	sgd	1.04E-05	-29.335
0.00124	torch_adamw	0.00011	-30.203

We have tested multiple sets of hyperparameters with varying ranges and prior distributions. Our hyperparameter tuning configuration input file is prepared in YAML format. Discrete hyperparameter values such as the number of epochs and the batch size are sampled from uniform distributions whereas continuous hyperparameters such as the learning rate and the gradient clip are sampled from normal distributions with/without log scale. The ranges of hyperparameters along with the prior distributions for sampling are both user-customizable.

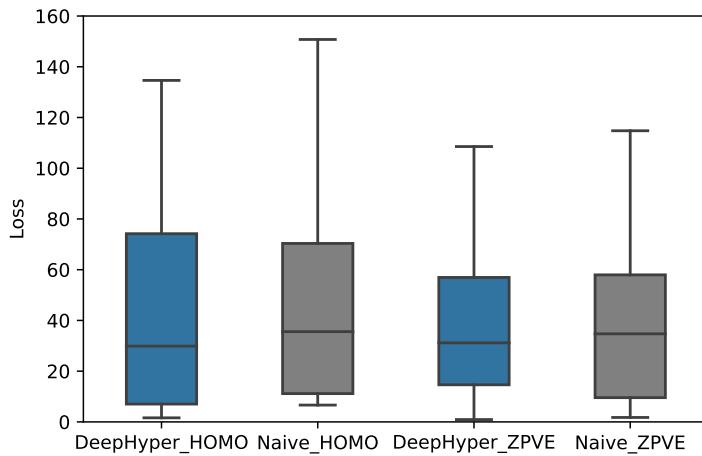
Once the hyperparameter configuration and the prior distributions are in place, DeepHyper can use multiple GPUs to perform hyperparameter tuning, taking full advantage of GPU parallelization. Next, all the optimization results will be saved and automatically logged to Weights and Biases. If the tuning step is interrupted, it can be resumed from the last saved checkpoint by specifying the `--resume` tag.

For the PhysNet model with HOMO and ZPVE as target properties, we compared hyperparameter tuning performance of DeepHyper with a naive algorithm that performs random selection of hyperparameters. Since DeepHyper utilizes the CBO algorithm to optimize hyperparameters, the target property values are used for decision making. For the naive algorithm, however, hyperparameters were randomly selected from the hyperparameter grid in Table 1. A total of 20 models were trained for 30 epochs with hyperparameters given by the two methods. The distributions of the losses (means squared error) are compared in Figure 1, and the metrics are summarized in Table 4.

*Key findings:* For the prediction of HOMO and ZPVE, DeepHyper yields better hyperparameter combinations, which accelerate convergence and provide optimal performance.

#### 4.2. AI model training

We trained PhysNet, SchNet, MPNN and MPNN-transformer (with attention mechanism) with HOMO and ZPVE as target properties from the QM9 dataset. The models were trained



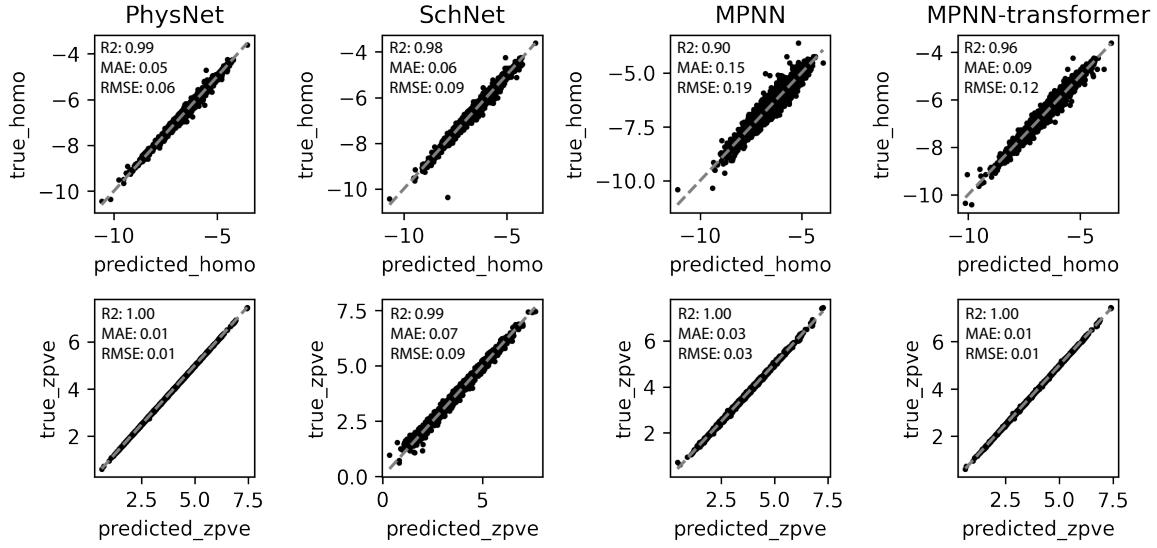
**Figure 1.** Comparison of the loss distributions of **PhysNet** with hyperparameters found by **DeepHyper** (blue) and a naive random selection algorithm (grey). Two outliers for the DeepHyper\_HOMO box were neglected to retain details.

**Table 4.** Performance of 20 models with hyperparameters found by **DeepHyper** and a naive random selection algorithm with **HOMO** and **ZPVE** as target properties. For both properties, the minimum loss and the standard deviation of loss are reported.

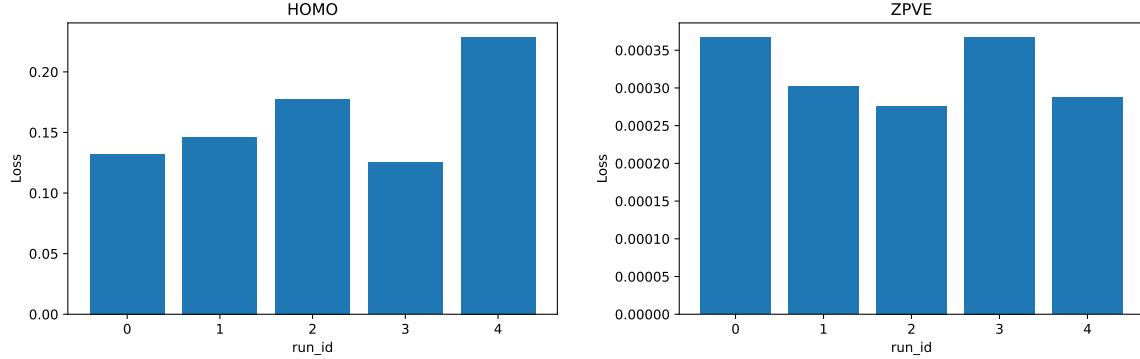
	HOMO		ZPVE	
	min_loss	std_loss	min_loss	std_loss
DeepHyper	1.604	74.871	0.923	31.771
Naive Algorithm	6.632	52.43	1.751	65.117

for 1,500 epochs to ensure convergence of validation loss. A new model is saved when the validation loss drops. The model training results are summarized in Figure 2. We found that **ZPVE** is an easier property to learn compared to **HOMO** for all four models, as indicated by a significantly lower loss. Moreover, the addition of attention layer in the **MPNN** model (**MPNN-transformer**) further lowers the mean absolute error (0.09 eV for **HOMO** and 0.01 eV for **ZPVE**) compared to the original **MPNN** model (0.15 eV for **HOMO** and 0.03 eV for **ZPVE**).

Model uncertainty quantification was performed for **PhysNet** model with **HOMO** and **ZPVE** as target properties. Five **PhysNet** models with randomly initialized weights were generated using the random seeds method. The optimal hyperparameter combinations found by **DeepHyper** in Section 4.1 were used. The models were trained for 100 epochs to achieve convergence of loss function. Figure 3 shows that **PhysNet** makes consistent predictions regardless of the random initial weights. The standard deviations of losses for the five models with **HOMO** and **ZPVE** as target properties are 0.0379 eV and 3.9646e-05 eV, respectively, and the mean absolute errors are comparable with those reported in the literature [27].



**Figure 2.** From left to right, model inference performance of PhysNet, SchNet, MPNN and MPNN-transformer with HOMO (top row) or ZPVE (bottom row) as the target property.



**Figure 3.** Model training performance of PhysNet with HOMO (left) and ZPVE (right) as target properties, initialized with 5 random seeds.

*Key findings:* Our suite of AI models provide state-of-the-art results. Novel features that we added to the models, such as attention to MPNN-transformer, further improve their performance. We have also demonstrated that hyperparameter optimization leads to stable, statistically robust AI predictions.

#### 4.3. Model improvement via modified node and edge embedding schemes

We modified the node and edge embedding schemes of CGCNN model from the original adjacency matrix format to an adjacency list format. There are two main advantages in using the adjacency list format. First, compared to the adjacency matrix format, it takes up less memory for loading, which speeds up model training. Second, the redundant information (zero paddings) in the representation is removed, resulting in

higher training accuracy and stability. As an example, CGCNN models with the two embedding schemes were trained on a subset of the hMOF database [28], which contains 5,000 randomly selected MOF structures along with their CO<sub>2</sub> working capacities at 2.5 bar. The model training results are shown in left panel of Figure 4. To smooth out local fluctuations, thirty point moving averaging was performed on both curves. We notice that CGCNN model with adjacency list format node and edge embeddings achieved faster convergence speed, higher training stability, and a lower mean absolute error (MAE) compared to the original CGCNN model.

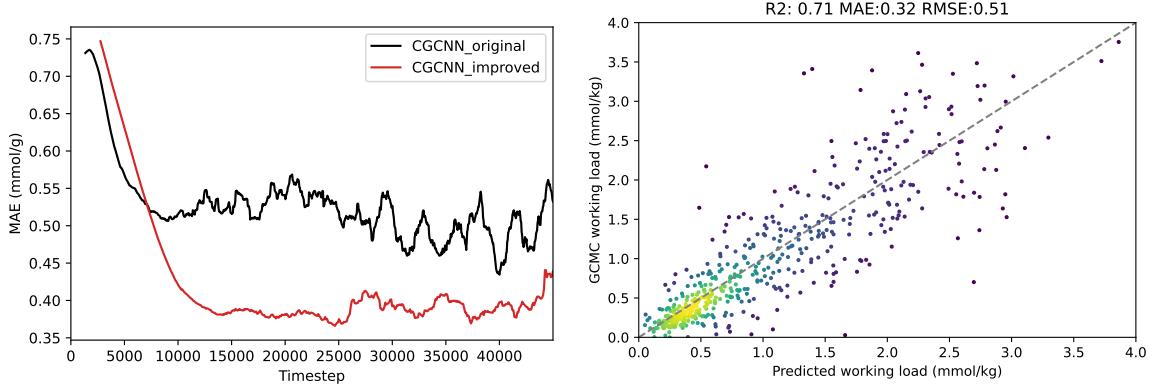
From the right panel of Figure 4, we show that the improved CGCNN model predicts CO<sub>2</sub> working capacity with a MAE of 0.32 mmol/g. To better understand the predictive performance of the improved CGCNN model, we benchmarked it against two recently proposed machine learning models for predicting CO<sub>2</sub> working capacity of MOFs, namely ALIGNN [29] and random forest regressor [30].

ALIGNN was trained on the entire hMOF dataset and predicts CO<sub>2</sub> working capacity at 2.5 bar with a MAE of 0.48 mmol/g, which is 50% higher loss value than our model. This could be because they trained the model on the entire hMOF dataset, whereas we randomly sampled around 5000 structures from the database for training. Note that ALIGNN uses both normal graph and line-graph (i.e., edges of normal graph are line-graph nodes while line-graph edges are interactions between line-graph nodes) for training and inference of working capacity prediction. For a normal graph, it uses physical and chemical features for atom (node) embedding, and distances between atoms as edge embedding; for a line graph, distances are line-graph node embedding while bond angle embedding is line-graph edge embedding. This scheme, however, can cause occasional CUDA memory issues and training batch size may have to be reduced (64 in [29]; 32 in our independent experiment), hence slower training. On the other hand our improved CGCNN model only takes in crystal structures as input (i.e., atom species and Cartesian coordinates) with larger batch size (256 in our model).

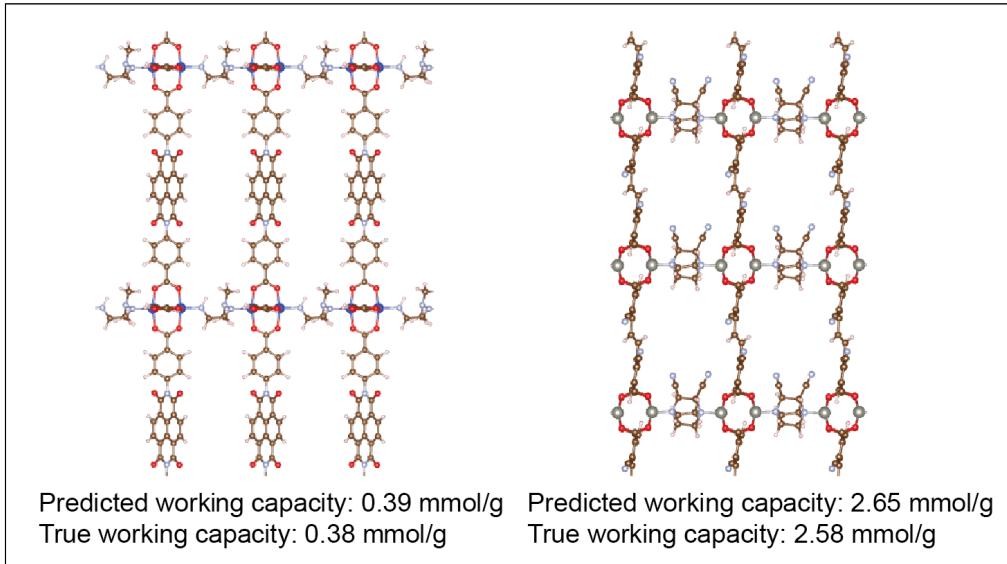
The random forest regression model takes in topological, structural, and word embeddings features as input. When trained on the entire hMOF dataset, it achieves an RMSE and R<sup>2</sup> score of roughly 0.65 and 0.95, respectively, for the prediction of CO<sub>2</sub> working capacity at 2.5 bar, which is 27% higher error value for RMSE than our model. Overall, the improved CGCNN model achieves competitive predictive performance compared to state-of-the-art machine learning models.

In other papers not using hMOF dataset and/or predicting other MOF properties, extensive physical and chemical featurizations were used [31][32], whereas our model learns the MOF information from only atom species and coordinates.

*Key findings:* Adopting adjacency list format node and edge embedding scheme improves the predictive capabilities of our improved CGCNN model. Using a test set of over 500 MOFs from the hMOF dataset, we have found that our improved CGCNN model provides state-of-the-art predictions for CO<sub>2</sub> working capacities at 2.5 bar.



**Figure 4.** Left panel, comparison of the original CGCNN model (black) with adjacency matrix format node and edge embedding schemes and the modified CGCNN model (red) with adjacency list format node and edge embedding schemes. Right panel, predictive performance of our improved CGCNN model on a test set (10% of our sampled 5000 structures) of 522 MOFs of the hMOF dataset. The lower and upper bound of both axes were constrained to 0 mmol/g and 4 mmol/g to reflect the typical working capacity range.

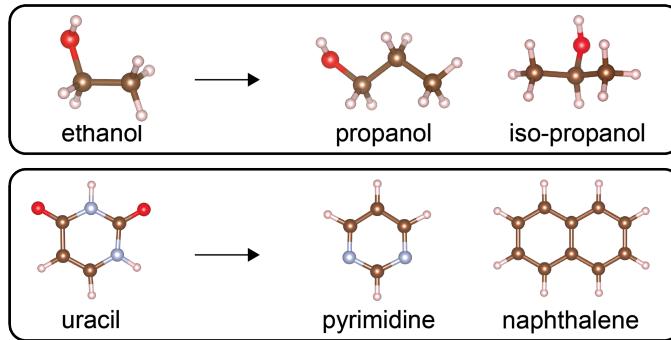


**Figure 5.** Sample MOF structures in the hMOF database along with the AI predicted and ground truth  $\text{CO}_2$  working capacities at 2.5 bar.

#### 4.4. Translational AI Applications

MD simulations of two sets of small molecules were performed to demonstrate the transferability of TorchMD-NET: from ethanol to *n*-propanol and *iso*-propanol and from uracil to pyrimidine and naphthalene.

In each set, the TorchMD-NET model trained with MD trajectories of the molecule on the left was used to perform MD simulations of the molecules on the right. The NVE ensemble was used, where the total number of particles and the simulation box volume is fixed and the total energy is conserved. For all molecules, the number of timesteps



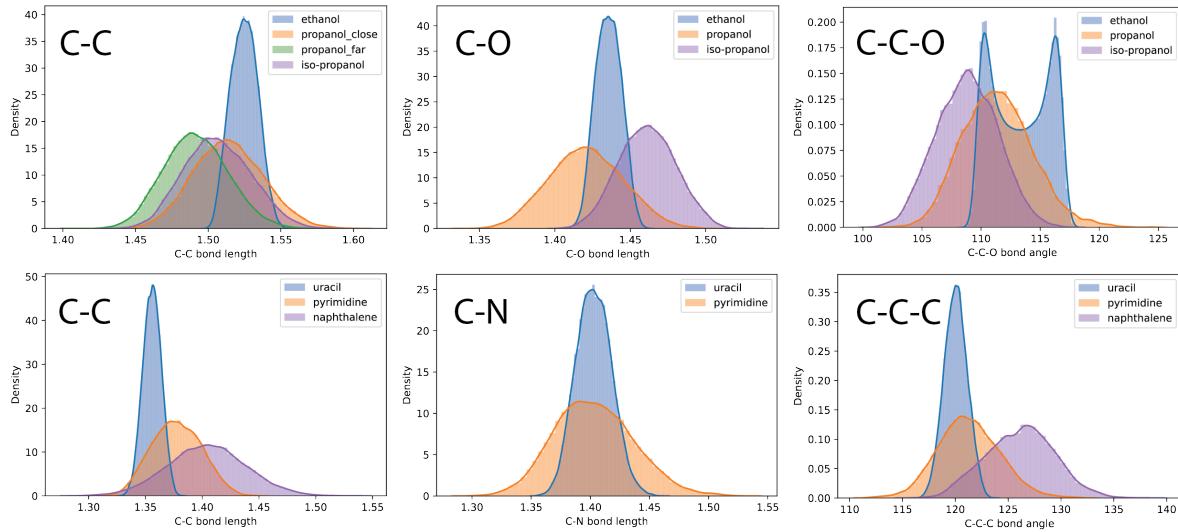
**Figure 6.** Example molecules used to demonstrate the transferability of TorchMD-NET. MD trajectories of molecules on the left are used to train the TorchMD-NET model, which is then used to perform MD simulations for the molecules on the right.

and the total simulation time were chosen to be 0.1 fs and 10 ps (100,000 timesteps), respectively. Figure 7 shows that the C-C and C-O bond length distributions of ethanol, *n*-propanol and *iso*-propanol have similar means, whereas the latter two have a larger spread. It is worth noting that for *n*-propanol, the length of the C-C bond closer to the O atom has a very similar distribution to that of ethanol, which is expected because their local environments are similar. For bond angles, The C-C-O bond angle distribution of ethanol exhibits two peaks, whereas the other two only have one peak. For uracil, pyrimidine and naphthalene, the C-C bond length distribution of naphthalene is shifted to a higher range compared to the other two, which may be due to the absence of N atoms in its ring structure. The C-N bond length distributions of uracil and pyrimidine have similar means, whereas the latter has a larger spread. Similarly, we observe comparable C-C-C bond angle distributions in uracil and pyrimidine, whereas the same distribution for naphthalene is shifted to a higher range, again an effect which may be due to the absence of N atoms in naphthalene's ring structure. The similarity and differences of bond length and angle distributions demonstrate that TorchMD-NET trained on one type of molecule is transferable to other similar molecules.

*Key findings:* We present a novel application of TorchMD-NET, in which this AI model was fine-tuned to describe a given small molecule by accurately predicting its potential energy and forces and perform NVE MD simulations, and then seamlessly used to describe other molecules with different structures, while still capturing physically realistic bond length and angle distributions.

#### 4.5. Interpretable Inference

**Model performance attribution.** By projecting the second last layer's high dimensional vector representation of graph neural network onto molecular structure and visualizing the projection, we can better understand the physical and chemical properties of the input data that affect predictions of our AI models. The top panels of Figure 8 present model interpretation results of how PhysNet model predicts HOMO

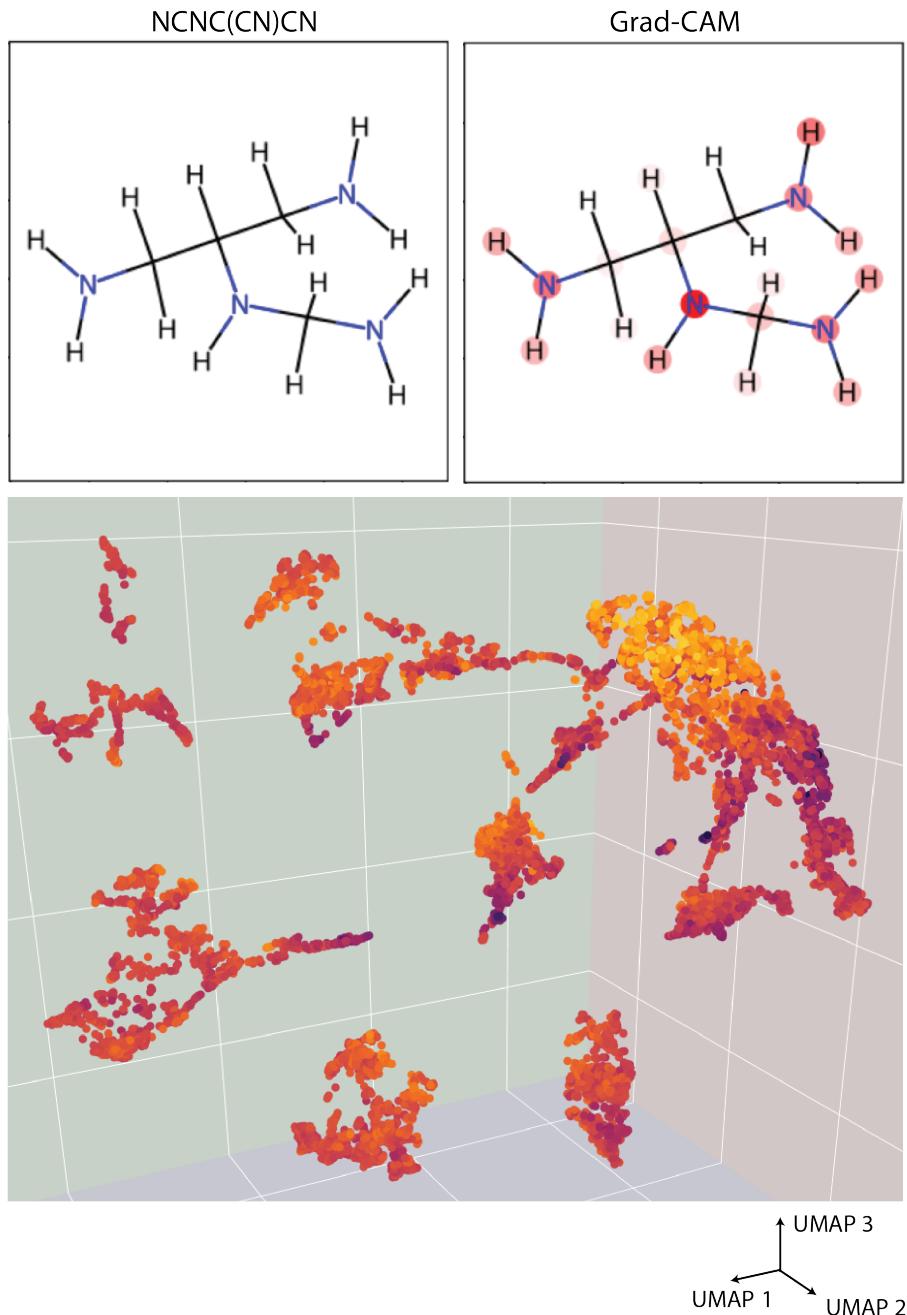


**Figure 7.** Top row, distributions of select bond lengths and angles in ethanol, *n*-propanol, and *iso*-propanol. Bottom row, distributions of select bond lengths and angles in uracil, pyrimidine, and naphthalene.

based on molecular structures via the Grad-CAM method. The N atoms and the H atoms connected to them are highlighted, possibly indicating that for **PhysNet**, these atoms carry more weight in the prediction of HOMO. We do not claim that explanations found by our deep learning model are the definite reasons for accurate prediction of QM properties such as HOMO or ZPVE, since these QM properties may not be simply determined by atomic species and coordinates. However, AI-explained visualization can help us better make sense of the patterns of complex molecular property predictions.

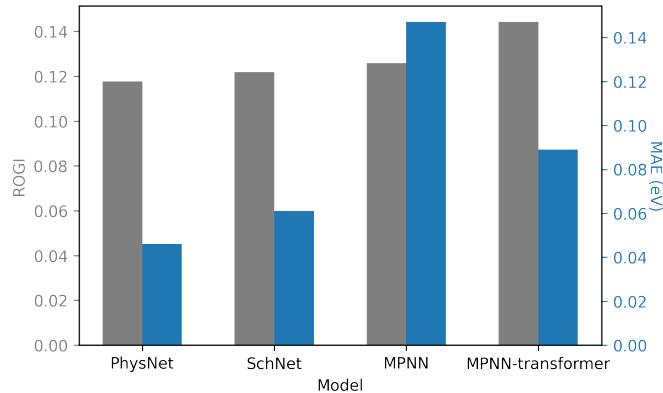
**Dimension Reduction** To reveal the correlation between model features and target properties, we applied uniform manifold approximation and projection (UMAP) dimension reduction technique to find the distribution of small molecules by projecting their high-dimensional structural/chemical data onto lower-dimension spaces. UMAP has been shown to achieve comparable or even better performance than other dimension reduction techniques such as principal component analysis (PCA) [33] and t-distributed stochastic neighbor embedding (tSNE) [34] on non-linear datasets [35]. Dimension reduction result for the **PhysNet** model with HOMO as target property is shown in the bottom panel of Figure 8, where each dot in the plot represents a molecule, color-coded based on its corresponding HOMO value. A 10% randomly selected subset (13.4k molecules) of the QM9 dataset was used to produce these results, which consists of stable small molecules composed of CHONF. From the scatter plot we know that molecules with similar HOMO values are clustered together and there is clear separation of molecules with low and high HOMO values. We present additional illustrative results in [Appendix B](#).

**Roughness of Molecular Property Landscape.** For molecular property prediction,



**Figure 8.** Top panels: exploration of the PhysNet model prediction using Grad-CAM method. Heavier atoms (N atoms) and the H atoms connected to them are highlighted in red, indicating their higher weight in HOMO prediction. Bottom panel: Uniform manifold approximation and projection (UMAP) dimension reduction results for the PhysNet model with HOMO as target property. Gold and purple dots represent molecules with high and low HOMO values, respectively.

the predictive performance of graph neural networks has shown to correlate to the roughness of molecular property landscape [36] [37] [38]. We adopted the recently proposed state-of-the-art roughness index (ROGI) [39] to measure how rough the HOMO and ZPVE landscapes are for PhysNet, SchNet, MPNN, MPNN-transformer. The calculation of molecular landscape roughness involves specifying a molecular representation and a distance metric. Molecular representations can be either learned by the graph neural network, with values extracted from the second last layer, or calculated based on molecular structures, as represented by SMILES strings or 3D Cartesian coordinates. The distance metrics are used to measure how different two molecular representations are. Example distance metrics include Tanimoto similarity, Euclidean distance, cityblock distance and cosine distance. To calculate ROGI values, learned molecular representation and one of the aforementioned distance metrics are used. Using Euclidean distance as the distance metric and HOMO as the target property, we show the ROGI values and mean absolute errors of four graph neural networks in Figure 9. We observe that a lower ROGI value in general corresponds to a lower mean absolute error, that is, higher predictive performance. We attribute this trend to the direct relation of a higher performing model to the smoothness of the resulting molecular property landscape. The exception is MPNN, which corresponds to a lower ROGI value despite a higher MAE as compared to the MPNN-transformer, which may be because the addition of transformer layers roughens the molecular property landscape while facilitating model training.



**Figure 9.** Roughness of the HOMO landscape (gray) and mean absolute error of model predictions (blue) for PhysNet, SchNet, MPNN and MPNN-transformer

*Key findings:* Our proposed approach brings together disparate interpretability AI tools to explore and make sense of AI model predictions, encompassing model performance attribution and scientific visualization; dimension reduction with UMAP to explore clustering of molecules with similar properties; and metrics such as the roughness index to quantify the predictive performance of our AI models for QM properties. These complementary tools provide valuable insights into the features and patterns of input data that are relevant for AI inference.

## 5. Conclusion

The rise of AI in the early 2010s was possible by a combination of elements, including disruptive technologies and computing approaches, as well as the desire to advance state-of-the-art practice through collaborative and friendly competitions in which high-quality datasets and AI models were freely shared. Similar approaches have been mirrored in science and engineering in recent years. These efforts are now being formalized through FAIR (findable, accessible, interoperable and reusable) initiatives [40, 41] in the context of scientific datasets [42], research software [43] and AI models [8, 44]. This study represents yet another significant step in this direction. We have assembled benchmark datasets, added novel features to state-of-the-art graph neural networks and transformer models, coupled them with robust libraries for hyperparameter tuning to improve their capabilities for scientific discovery, and developed and adapted a set of visualization and interpretability tools to make sense of the AI predictions. All these elements are unified within a single computational framework that has been deployed and extensively tested on leadership-class, high-performance computing platforms. Researchers using this computational framework will be able to conduct scientific discovery combining state-of-the-art AI models with datasets that are coupled with advanced supercomputing platforms. We expect that this approach will catalyze the sharing of AI knowledge and tools in the context of molecular and crystal property prediction applications.

## 6. Acknowledgments

This work was supported by the FAIR Data program and the Braid project of the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, under contract number DE-AC02-06CH11357. It used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. This work was supported by Laboratory Directed Research and Development (LDRD) funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357. This research used the Delta advanced computing and data resource which is supported by the National Science Foundation (award OAC 2005572) and the State of Illinois. Delta is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. We thank Prasanna Balaprakash and the [DeepHyper](#) team for their expert support and guidance as we coupled their library into our computational AI framework.

## Code availability

The AI models `PhysNet` [45] and `MPNN_transformer` [46] presented in this study are freely available at the Data and Learning Hub for Science [47, 48].

## ORCID IDs

Santanu Chaudhuri [0000-0002-4328-2947](#)

Eliu Huerta [0000-0002-9682-3604](#)

Hyun Park [0000-0001-5550-5610](#)

Emad Tajkhorshid [0000-0001-8434-1010](#)

Ruijie Zhu [0000-0001-9316-7245](#)

## References

- [1] Schütt K T, Sauceda H E, Kindermans P J, Tkatchenko A and Müller K R 2018 *The Journal of Chemical Physics* **148** 241722
- [2] Unke O T and Meuwly M 2019 *Journal of chemical theory and computation* **15** 3678–3693
- [3] Thölke P and De Fabritiis G 2022 *arXiv preprint arXiv:2202.02541*
- [4] Klicpera J, Groß J and Günnemann S 2020 *arXiv preprint arXiv:2003.03123*
- [5] Xie T and Grossman J C 2018 *Physical review letters* **120** 145301
- [6] Liu M, Luo Y, Wang L, Xie Y, Yuan H, Gui S, Yu H, Xu Z, Zhang J, Liu Y, Yan K, Liu H, Fu C, Oztekin B M, Zhang X and Ji S 2021 *Journal of Machine Learning Research* **22** 1–9 URL <http://jmlr.org/papers/v22/21-0343.html>
- [7] Fung V, Zhang J, Juarez E and Sumpter B G 2021 *npj Computational Materials* **7** 1–8
- [8] Ravi N, Chaturvedi P, Huerta E A, Liu Z, Chard R, Scourtas A, Schmidt K J, Chard K, Blaiszik B and Foster I 2022 *Scientific Data* **9** 657 ISSN 2052-4463 URL <https://doi.org/10.1038/s41597-022-01712-9>
- [9] Huerta E A, Khan A, Davis E, Bushell C, Gropp W D, Katz D S, Kindratenko V, Koric S, Kramer W T C, McGinty B, McHenry K and Saxton A 2020 *Journal of Big Data* **7** 88 (*Preprint 2003.08394*)
- [10] Huerta E A, Khan A, Huang X, Tian M, Levental M, Chard R, Wei W, Heflin M, Katz D S, Kindratenko V, Mu D, Blaiszik B and Foster I 2021 *Nature Astronomy* **5** 1062–1068 (*Preprint 2012.08545*)
- [11] Balaprakash P, Salim M, Uram T D, Vishwanath V and Wild S M 2018 Deephyper: Asynchronous hyperparameter search for deep neural networks *2018 IEEE 25th international conference on high performance computing (HiPC)* (IEEE) pp 42–51
- [12] Ruddigkeit L, Van Deursen R, Blum L C and Reymond J L 2012 *Journal of chemical information and modeling* **52** 2864–2875
- [13] Wilmer C E, Farha O K, Bae Y S, Hupp J T and Snurr R Q 2012 *Energy & Environmental Science* **5** 9849–9856
- [14] Chmiela S, Tkatchenko A, Sauceda H E, Poltavsky I, Schütt K T and Müller K R 2017 *Science advances* **3** e1603015
- [15] Biewald L 2020 Experiment tracking with weights and biases software available from wandb.com URL <https://www.wandb.com/>
- [16] Gasteiger J, Becker F and Günnemann S 2021 *Advances in Neural Information Processing Systems* **34** 6790–6802
- [17] Choudhary K and DeCost B 2021 *npj Computational Materials* **7** 1–8
- [18] Chen C, Ye W, Zuo Y, Zheng C and Ong S P 2019 *Chemistry of Materials* **31** 3564–3572
- [19] Larsen A H, Mortensen J J, Blomqvist J, Castelli I E, Christensen R, Dułak M, Friis J, Groves M N, Hammer B, Hargus C et al. 2017 *Journal of Physics: Condensed Matter* **29** 273002
- [20] Wang M, Zheng D, Ye Z, Gan Q, Li M, Song X, Zhou J, Ma C, Yu L, Gai Y et al. 2019 *arXiv preprint arXiv:1909.01315*
- [21] Fey M and Lenssen J E 2019 Fast graph representation learning with PyTorch Geometric *ICLR Workshop on Representation Learning on Graphs and Manifolds*

- [22] Leow Y Y, Laurent T and Bresson X 2019 Graphsne: A visualization technique for graph-structured data *ICLR Workshop on Representation Learning on Graphs and Manifolds*
- [23] Gelman S, Fahlberg S A, Heinzelman P, Romero P A and Gitter A 2021 *Proceedings of the National Academy of Sciences* **118** e2104878118
- [24] Mnih V, Kavukcuoglu K, Silver D, Rusu A A, Veness J, Bellemare M G, Graves A, Riedmiller M, Fidjeland A K, Ostrovski G et al. 2015 *nature* **518** 529–533
- [25] Pope P E, Kolouri S, Rostami M, Martin C E and Hoffmann H 2019 Explainability methods for graph convolutional neural networks *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp 10772–10781
- [26] Papers with Code - Gradient Clipping Explained <https://paperswithcode.com/method/gradient-clipping> [Online; accessed 2022-11-21]
- [27] Glavatskikh M, Leguy J, Hunault G, Cauchy T and Da Mota B 2019 *Journal of cheminformatics* **11** 1–15
- [28] Bucior B J, Rosen A S, Haranczyk M, Yao Z, Ziebel M E, Farha O K, Hupp J T, Siepmann J I, Aspuru-Guzik A and Snurr R Q 2019 *Crystal Growth & Design* **19** 6682–6697
- [29] Choudhary K, Yildirim T, Siderius D W, Kusne A G, McDannald A and Ortiz-Montalvo D L 2022 *Computational Materials Science* **210** 111388
- [30] Krishnapriyan A S, Montoya J, Haranczyk M, Hummelshøj J and Morozov D 2021 *Scientific reports* **11** 1–11
- [31] Burner J, Schwiedrzik L, Krykunov M, Luo J, Boyd P G and Woo T K 2020 *The Journal of Physical Chemistry C* **124** 27996–28005
- [32] Moosavi S M, Novotny B Á, Ongari D, Moubarak E, Asgari M, Kadioglu Ö, Charalambous C, Guerrero A, Farmahini A H, Sarkisov L et al. 2022
- [33] Jolliffe I T and Cadima J 2016 *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **374** 20150202
- [34] Van der Maaten L and Hinton G 2008 *Journal of machine learning research* **9**
- [35] Wang Y, Huang H, Rudin C and Shaposhnik Y 2021 *J. Mach. Learn. Res.* **22** 1–73
- [36] Peltason L and Bajorath J 2007 *Journal of medicinal chemistry* **50** 5571–5578
- [37] Guha R and Van Drie J H 2008 *Journal of chemical information and modeling* **48** 646–658
- [38] Golbraikh A, Muratov E, Fourches D and Tropsha A 2014 *Journal of chemical information and modeling* **54** 1–4
- [39] Aldeghi M, Graff D E, Frey N, Morrone J A, Pyzer-Knapp E O, Jordan K E and Coley C W 2022 *Journal of Chemical Information and Modeling* **62** 4660–4671
- [40] Wilkinson M D, Sansone S A, Schultes E, Doorn P, da Silva Santos L O B and Dumontier M 2018 *Scientific Data* **5** 180118 URL <https://doi.org/10.1038/sdata.2018.118>
- [41] Wilkinson M D, Dumontier M, Aalbersberg I J, Appleton G, Axton M, Baak A, Blomberg N, Boiten J W, da Silva Santos L B, Bourne P E, Bouwman J, Brookes A J, Clark T, Crosas M, Dillo I, Dumon O, Edmunds S, Evelo C T, Finkers R, Gonzalez-Beltran A, Gray A J G, Groth P, Goble C, Grethe J S, Heringa J, 't Hoen P A C, Hooft R, Kuhn T, Kok R, Kok J, Lusher S J, Martone M E, Mons A, Packer A L, Persson B, Rocca-Serra P, Roos M, van Schaik R, Sansone S A, Schultes E, Sengstag T, Slater T, Strawn G, Swertz M A, Thompson M, van der Lei J, van Mulligen E, Velterop J, Waagmeester A, Wittenburg P, Wolstencroft K, Zhao J and Mons B 2016 *Sci. Data* **3** 160018
- [42] Chen Y, Huerta E A, Duarte J, Harris P, Katz D S, Neubauer M S, Diaz D, Mokhtar F, Kansal R, Park S E, Kindratenko V V, Zhao Z and Rusack R 2022 *Scientific Data* **9** 31 (Preprint [2108.02214](https://arxiv.org/abs/2108.02214))
- [43] Barker M, Chue Hong N, Katz D S, Lamprecht A L, Martinez Ortiz C, Psomopoulos F, Harrow J, Castro L, Gruenpeter M, Martinez P and Honeyman T 2022 *Scientific Data* **9**
- [44] Duarte J, Li H, Roy A, Zhu R, Huerta E A, Diaz D, Harris P, Kansal R, Katz D S, Kavoori I H, Kindratenko V V, Mokhtar F, Neubauer M S, Eon Park S, Quinnan M, Rusack R and Zhao Z 2022 *arXiv e-prints* arXiv:2212.05081 (Preprint [2212.05081](https://arxiv.org/abs/2212.05081))

- [45] Park, Hyun and Zhu, Ruijie and Huerta, EA 2022 PhysNet for molecular dynamics applications in leadership class supercomputers. The Data and Learning Hub for Science <https://doi.org/10.26311/b6x8-4621>
- [46] Park, Hyun and Zhu, Ruijie and Huerta, EA 2022 MPNN\_transformer for molecular dynamics applications in leadership class supercomputers. The Data and Learning Hub for Science <https://doi.org/10.26311/3pm2-am44>
- [47] Chard R, Li Z, Chard K, Ward L, Babuji Y, Woodard A, Tuecke S, Blaiszik B, Franklin M J and Foster I 2019 DLHub: Model and data serving for science *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* pp 283–292
- [48] Li Z, Chard R, Ward L, Chard K, Skluzacek T J, Babuji Y, Woodard A, Tuecke S, Blaiszik B, Franklin M J and Foster I 2021 *J. Parallel. Distrib. Comput.* **147** 64 ISSN 0743-7315

## Appendix A. Hyperparameter tuning results of PhysNet with HOMO as target property

**Table A1.** Top 10 DeepHyper hyperparameter combinations for PhysNet with HOMO as target property.

agb	amp	batch_size	gradient_clip
3	TRUE	235	1.36E-01
1	TRUE	349	1.10E+00
14	FALSE	130	5.40E-05
3	FALSE	159	1.79E-02
5	TRUE	404	8.24E-02
4	TRUE	460	7.21E-02
13	TRUE	160	3.42E-05
4	FALSE	147	6.16E-02
7	TRUE	163	1.90E-01
1	TRUE	258	3.58E-02

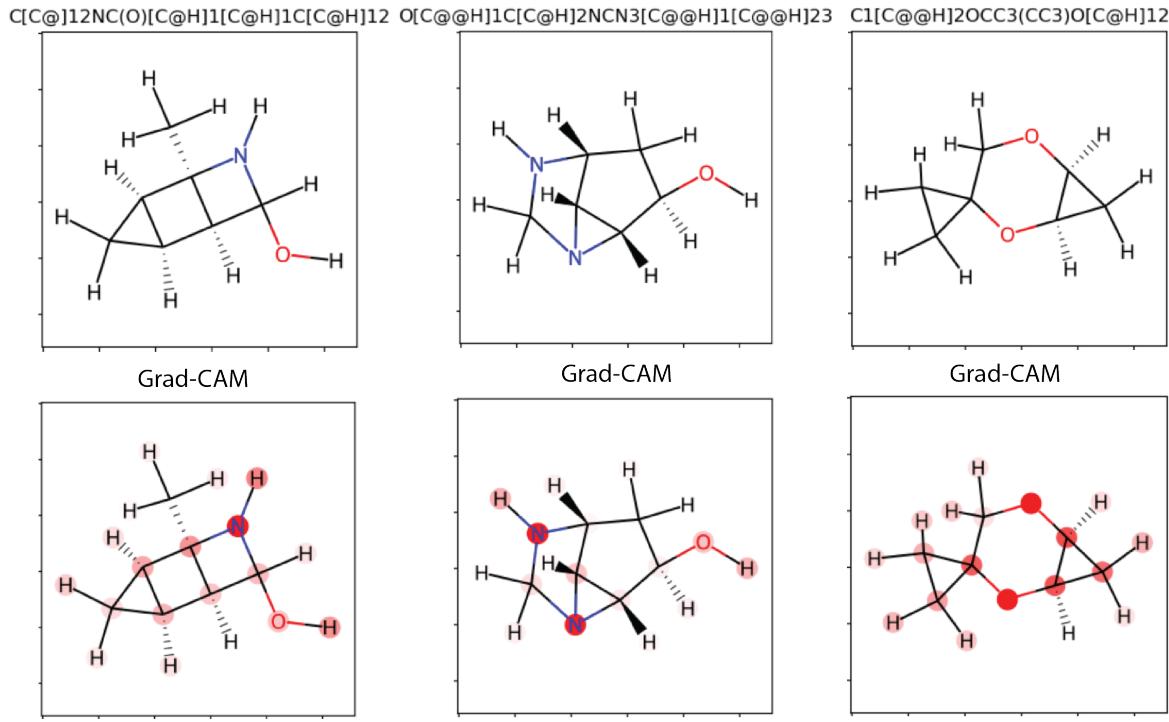
**Table A2.** As Table A1 for the rest of parameters optimized through DeepHyper.

learning_rate	optimizer	weight_decay	objective
1.15E-03	sgd	2.94E-05	-1.604
8.69E-04	sgd	2.30E-06	-2.454
1.49E-04	lamb	1.17E-03	-2.976
5.64E-01	lamb	2.09E-04	-3.323
1.38E-03	sgd	4.83E-05	-6.474
2.33E-02	sgd	2.84E-04	-7.214
4.61E-04	lamb	1.74E-04	-12.494
5.99E-04	torch adamw	1.59E-03	-14.5111
5.26E-01	lamb	1.44E-04	-18.0449
6.95E-01	lamb	8.62E-05	-29.505

## Appendix B. Examples of model explainability features

We present results to complement the interpretable AI analysis presented in Section 4.5. Figure B1 illustrates what information AI models may extract from input data to make predictions that are consistent with state-of-the-art knowledge on QM properties.

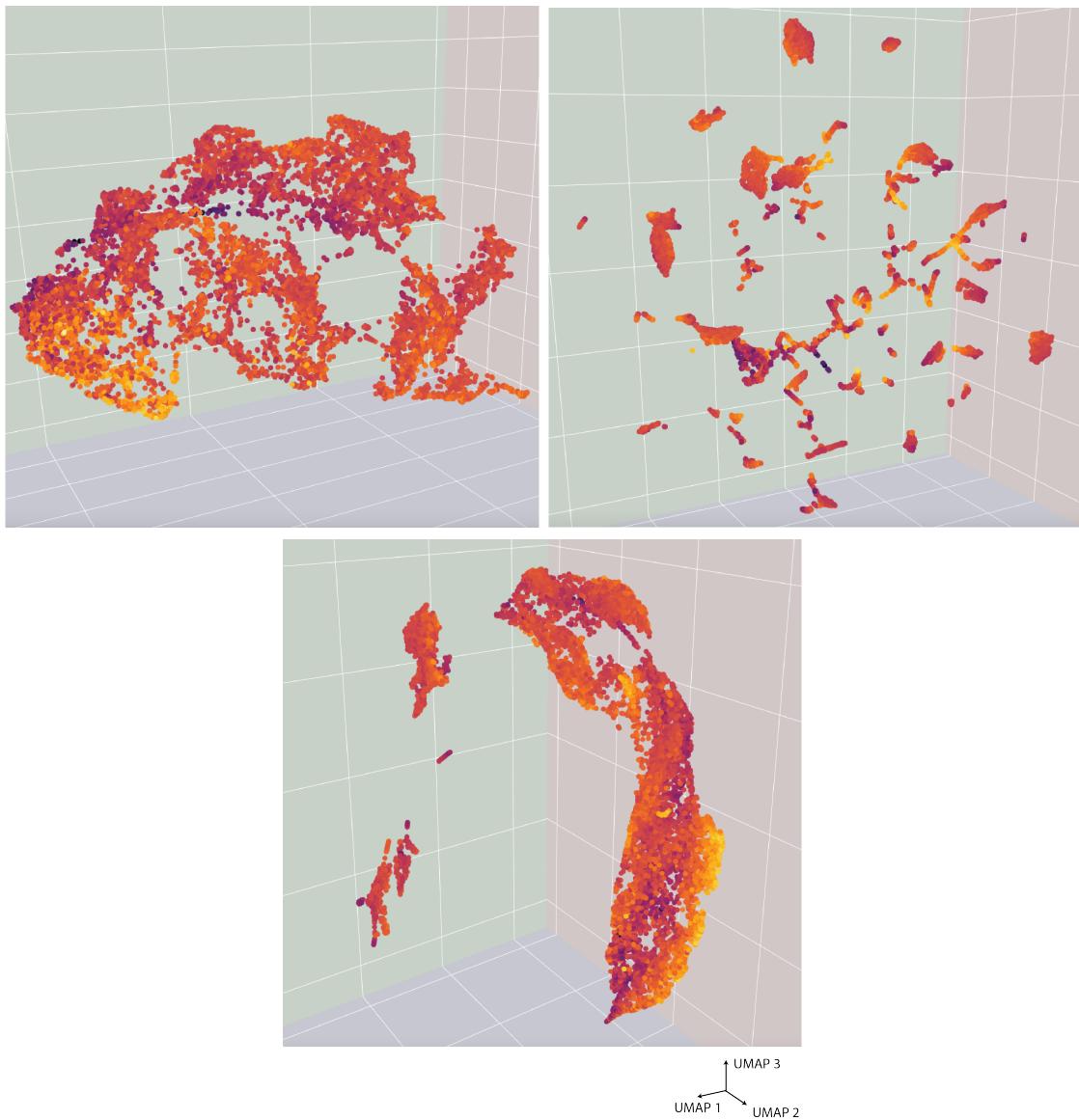
### Appendix B.1. Grad-CAM interpretation



**Figure B1.** Molecular graphs (top) and Grad-CAM interpretations (bottom) of PhysNet with HOMO as target property for three example molecules. The N atoms and the H atoms attached to them are highlighted in red, indicating their higher weight in model predictions.

### Appendix B.2. UMAP interpretation

Figure B2 shows that we can turn our AI predictors into feature extractors to explore clustering of molecules with similar properties.



**Figure B2.** UMAP dimension reduction results for **SchNet** (top left), **MPNN** (top right) and **MPNN-transformer** (bottom) with HOMO as target property. A randomly selected 10% subset (13.4k molecules) of the QM9 dataset was used for analysis, which consists of stable small organic molecules composed of CHONF.

## Research Article

# English-Chinese Machine Translation Based on Transfer Learning and Chinese-English Corpus

Bo Xu 

School of Foreign Studies, Huanggang Normal University, Huanggang 438000, China

Correspondence should be addressed to Bo Xu; [xu\\_bo@hgnu.edu.cn](mailto:xu_bo@hgnu.edu.cn)

Received 8 June 2022; Revised 29 August 2022; Accepted 3 September 2022; Published 27 September 2022

Academic Editor: Jun Ye

Copyright © 2022 Bo Xu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes an English-Chinese machine translation research method based on transfer learning. First, it expounds the theory of neural machine translation and transfer learning and related technologies. Neural machine translation is discussed, the advantages and disadvantages of various models are introduced, and the transformer neural machine translation model framework is selected. For low-resource Chinese-English parallel corpus and Tibetan-Chinese parallel corpus, 30 million Chinese-English parallel corpora, 100,000 Chinese-English low-resource parallel corpora, and 100,000 Tibetan-Chinese parallel corpora were used to pretrain the transformer machine translation architecture. The decoders are all composed of 6 identical hidden layers, the initialization of the model parameters is done by the transformer uniform distribution, and the model training uses Adam as the optimizer. In the model transfer part, the parameters with the better effect of the pretrained model are transferred to the low-resource Chinese-English and Tibetan-Chinese machine translation model training, so as to achieve the purpose of knowledge transfer. The results show that the model transfer learning of low-resource Chinese-English parallel corpus improves the translation system's translation by 3.97 BLEU values compared with the translation system without transfer learning at 0.34 BLEU values. Model transfer learning on low-resource Tibetan-Chinese parallel corpus increases the BLEU value by 2.64 BLEU compared to the translation system without transfer learning. The neural machine translation system that uses BPE technology for preprocessing plus model transfer learning is compared to the translation system that only performs transfer learning and shows an improved 0.26 BLEU value. It is verified that the transfer learning method proposed in this paper has a certain improvement in the effect of low-resource Chinese-English and Tibetan-Chinese neural machine translation models.

## 1. Introduction

As an essential element of human communication, with the development of time and international trade, people from all over the world communicate and cooperate more, people in more countries have more relationships, and the need for seamless communication and understanding has become very important. Using machine translation technology to solve language barriers is a valuable tool for solving human-to-human communication problems, and translators have always been the focus of attention among scientists. In today's age of intellectual property, machine translation has become possible, thanks to advances in software technology, the emergence of new algorithms, and improvements in computer performance. Machine translation has been used extensively in modern translation work, and its role and

impact are unpredictable. Some experts even predict that it will replace human translation in the future. Google online translation is a machine translation tool that can indeed help translators solve certain problems. Although it plays a different role in the translation of different texts, it still needs to be edited manually to varying degrees. In order to improve the quality and efficiency of machine translation and reduce the involvement of human translation, this paper focuses on English-Chinese translation and proposes a more efficient machine translation model.

## 2. Related Works

Song, Q. and others said that China's research in the field of machine translation began in 1957. It is the fourth country in the world to start machine translation research after the

United States, Germany, and the Soviet Union. In 1958, the first machine translation experiment was carried out on domestic 104 large general-purpose digital computers, successfully translating 20 different types of Russian sentences into Chinese [1]. Later, the Harbin Institute of Technology, the China Institute of Science and Information Technology, the South China Institute of Technology, and other organizations also set up the machine. Translate research teams and conduct research on English-Chinese or Russian-Chinese translators. Zhang, X, and others said that the phrases based on the Tibetan-Chinese translator are based on the features of Tibetan morphology and grammar. This article focuses on the use of Tibetan coding conversions and Tibetan automated word distribution in the system. In addition, this article contains important requests and guidelines for research and translation technology related to China, including the integration of the Tibetan Corps and the automatic language distribution of both equal languages [2]. Forty and others claim that “research on the Tibetan-Chinese neural network translator” is the first time that a neural translator has been used for the Tibetan-Chinese translator area [3]. Jin and others said the work used end-to-end models based on cyclic neural networks and monitoring networks, and that migration education was used to start modeling. Good for solving the problem of insufficient data for small models. In-depth case study [4], Ren et al. said that the model method has an improvement of 3 BLEU values compared to phrase-based Tibetan-Chinese machine translation in practical experiments, and this work is indispensable for the study of low-resource neural machine translation. It can be seen from the abovementioned literature that the research units of low-resource neural machine translation in China are mainly concentrated in universities including Northwest University for Nationalities, Tibet University, Harbin Institute of Technology, Inner Mongolia University of Technology, and other universities [5]. Xia and others said that in addition, some state organs and enterprises are also actively developing such software and providing relevant services. For example, China National Language Translation Bureau and Yayı Network Technology Co., Ltd. provide machine translation services between Tibetan-Chinese, Mongolian-Chinese, Uyghur-Chinese, and other language pairs [6]. Ming, N. et al. said that although these system services can temporarily alleviate the society's demand for machine translation for low-resource languages, these systems are all based on statistical models, compared to the English-Chinese provided by NetEase, Baidu, Google, Sogou, and other enterprises. English-German and other online translation services, there is still a big gap in translation quality [7]. He and others said that how to use a neural network to improve the translation quality of low resource language pairs and minimize this gap is the common concern and focus of current machine translation researchers related to low resource language [8]. The main disadvantage of the academics' work is that the network has already adapted to the overall input of 300-1000-dimensional vectors before it starts producing outputs. Therefore, some scholars have proposed the so-called attention mechanism. Ji, B. et al said that the attention mechanism

gives the network the ability to reconsider all input words and use this information when generating new words [9]. The previous architecture is redesigned with a convolutional neural network (CNN), which processes all input words together, so it makes the training and reasoning process faster. That same year, Google subversively proposed a neural translator model that left the entire cyclic neural network and convolutional neural network. Lin, L., and others said that the model would also use “encoder-decoder” as the basis for the model. In the structure, multiple head listening methods and feed-forward neural networks are used to design the encoder and decoder structure. The model has achieved impressive results in working as a translator for several languages. One way is to plan for the integration of language structures (LMs) that are learned in the NMT system, that is, only speech data ( $s$ ). Experimental results show that integration of monolingual corpus can improve translation problems (Turkish-English) and translation problems (Chinese writing in English) [10]. The principle of English-Chinese machine translation is shown in Figure 1.

### 3. Method

As one of the research centers in natural language processing technology, translation aims to enable computers to correctly understand and translate natural language like people. The development of machine translation technology has been closely associated with the development of computer technology, information theory, linguistics, and other disciplines. It aims to transform data into information that is relevant through communication and decision-making. Among them, translators play an important role in translation. This is a research analysis of how to quickly identify a translator model with high accuracy and robustness [11]. The basic framework of machine translation is shown in Figure 2.

Generally, machine translation can be regarded as the transformation from one sequence to another. Machine translation is a widely recognized and useful example of sequence-to-sequence models, and allows us to demonstrate the difficulties encountered in trying to solve these problems using many intuitive examples. The encoder sequentially encodes the text and deletes the language information in the distribution representation, and then the decoder converts the information representation into a presentation in other languages, as shown in Figure 3.

Figure 3 shows the relationship between the encoder and decoder. First, through the encoder, the source language sequence “ $x_1, x_2, \dots, x_T$ ” is encoded by the encoder to generate a vector  $C$  representation, and then the vector is sent to the decoder as an input, and the decoder decodes this vector into the target language sequence [12]. During the target language, sequence generation is performed word by word, when a certain word is generated, it depends on the historical information of the previously generated target language until the end of the sentence is generated. Recurrent neural network (RNN): a recurrent neural network that takes sequence data as input, performs recursion in the evolution direction of the sequence and connects all nodes in

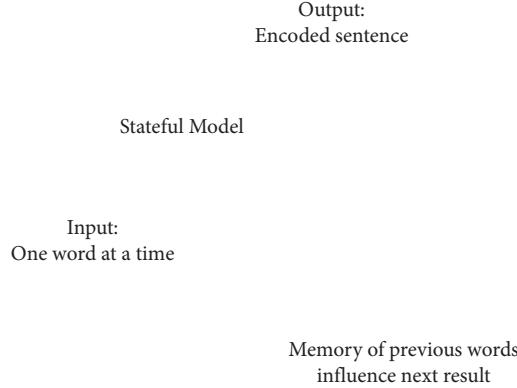


FIGURE 1: Principles of English-Chinese translation.

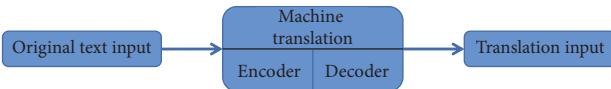


FIGURE 2: Basic framework of machine translation.

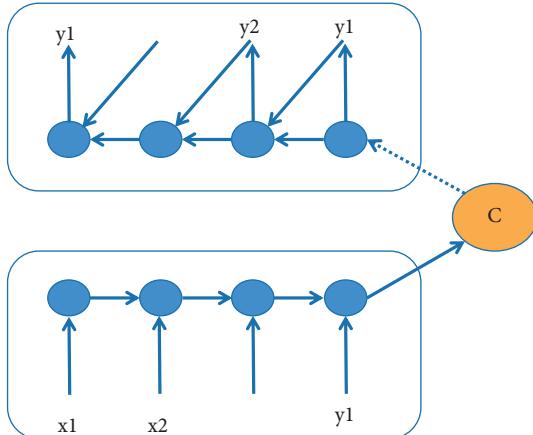


FIGURE 3: Example of encoder and decoder.

a chain. Cyclic neural network is mainly used to process sequence data, especially for variable length sequence data. Its core part is a directed graph. Chained elements in directed graph expansion are called recurrent units [13, 14]. RNN can be thought of as more than one application on the same neural network, and each neural network will transfer data to another location. Where  $x = \{x_1, x_2, \dots, x_t\}$  represent different data lengths. At point  $t$ , the hidden state  $h_t$  is changed by the following formula, as shown in the following formula:

$$h_t = f(h_{t-1}, x_t). \quad (1)$$

$f$  is a bugless operation,  $U$  is the weight matrix input to the hidden layer,  $V$  is the weight matrix from the hidden layer to the output layer,  $y$  is the target sequence to be achieved by the model,  $L$  is the loss function, and  $W$  is the weight matrix from the hidden layer to the hidden layer. The

time series  $t$  is in the range  $[1, t]$  and the input  $x$  is mapped to the output  $o$  by a recurrent neural network. The entire network is transformed by the following model, as shown in the following formula:

$$\begin{aligned} a_t &= Eh_{t-1} + Ux_t + b, \\ h_t &= \tanh(a_t), \\ o_t &= Vh_t + c, \\ \hat{y} &= \text{soft max}(o_t). \end{aligned} \quad (2)$$

The cyclic neural network unifies the length of the input vector of the input sequence of different lengths, and the same parameters and transformation energy can be used at any time point, which is required for operation files of different lengths. In addition, RNN can theoretically capture any precedent the idea of RNN is to use serialized information. In traditional neural networks, we assume that all inputs and outputs are independent of each other, but for many tasks, this assumption is problematic. For example, if you want to predict the next word in a sentence, you need to know which words come before it. LSTM differs from RNN. RNN can only have short memory due to gradient loss. The long-short-term memory network combines short-term memory with long-term memory by introducing gate control, which solves the problem of gradient disappearance to a certain extent. Short-term memory networks ... through directional control." is grammatically unclear. Please rephrase the sentence for clarity and correctness." LSTM is composed of three gate control units, namely, input gate, forget gate, and output gate. The input gate controls the input of the network, the forget gate controls the memory unit, and the output gate controls the output of the network [15]. The memory information at time  $t$  is used to save important information. Just like a record book, it saves the knowledge points learned in the past [16]. To control the content of forgetting the cell state of the previous layer, use sigmoid as the activation function,  $X_t$  of this sequence as the input, and then according to the  $h_{t-1}$  of the previous sequence, get the content of the cell state of the previous layer that needs to be removed and which needs to be retained. It should be noted that the input is in the form of a vector. We expect that the output value of the forget gate is mostly 0 or 1, that is, each value in the vector is completely forgotten or completely reserved, so we choose the sigmoid function as the activation function (6) shows.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f). \quad (3)$$

The input gate determines how much of the input  $x_t$  of the current network is retained to the current time  $c_t$ . This process is divided into two steps, the first is to use the input gate containing the sigmoid layer to decide what new information is added to the cell state; after determining the new data to be added, it is necessary to convert the new data into a format that can be added to the cell state. Thus, the second step is to use the tanh function to create a new candidate vector, as shown in the following formula:

$$\begin{aligned} i_t &= \sigma(W_f * [h_{t-1}, x_t] + b_f) \\ \tilde{C}_t &= \tanh(W_C * [h_{t-1}, x_t] + b_C) \end{aligned} \quad (4)$$

After the data are checked by the port and the port input, the state of the Ct-1 cell can be adjusted to Ct. As shown in the example below, where  $f_t \times C_{t-1}$  represents the information you want to delete and  $i_t \times C_t$  represents the new information, as shown in the following formula:

$$C_t = f_t * C_{t-1+i_t} * \tilde{C}_t. \quad (5)$$

How much of the control panel Ct is sent to the current output value ht to LSTM. That is, selectively release the contents of the cell state preservation. Like the updated two parts of the front door, the output gate also needs to use the sigmoid activation function to determine which information needs to be output. The cell state is processed through the tanh layer, multiply the two to get the information we want to output. We then use the tanh activation function to make the contents of the cell state and divide it into two sections to get what we want to release, as shown in the following formula:

$$\begin{aligned} ot &= \sigma(W_o * [ht - 1, xt] + bo), \\ ht &= (ot * \tanh(Ct)). \end{aligned} \quad (6)$$

GRU (gated recurrent unit) model is a type of RNN model. Like LSTM, it can intercept relationships with long-distance connections and reduce the likelihood of disappearing or breaking. At the same time, the structure and calculation are simpler than LSTM. GRU merges the forget gate and output gate into an “update gate,” which has a very good effect. Therefore, it is also a network structure of very manifold at present. To solve the problem of gradient vanishing and gradient explosion, the method and structure are shown in the following formula:

$$\begin{aligned} Zt &= \sigma(W_z * [ht - 1, xt]), \\ rt &= \sigma(W_z * [ht - 1, xt]), \\ ht &= \tanh(W * [rt * ht - 1, xt]), \\ ht &= (1 - Zt) * ht - 1 + Zt * \tilde{h}_t, \end{aligned} \quad (7)$$

where rt represents a gateway reset, which is used to determine the level of forgetfulness of previous data. Zt means to change the door. The update gate acts like the forget gate and input gate in LSTM. It determines, which data to forget and which new data to add to the neural structure. Each word will be represented as a real vector. This corresponds to a representation model of words. This section mainly introduces the difference between the traditional word representation model and the word representation model based on a real number vector. One hot coding is a traditional word representation method. A hot coding represents a word as a 0-1 vector of uppercase letters, where only the corresponding product for the word is 1, and all other objects are zero. For example, suppose a dictionary contains 10,000 words and numbers. Then each word can be represented as a 10k dimensional one-hot vector. Using Python

is an a-explanatory, object-oriented, dynamic data type advanced programming language to solve problems. Only the dimension corresponding to the number is 1, and the other dimensions are 0. The advantage of one hot coding is that the form is simple and easy to calculate, and this representation has a good correspondence with the dictionary, so each code can be interpreted. However, one hot coding regards words as mutually orthogonal vectors. This results in no correlation between all words. Single-thermal encoding is often used to handle features that do not have size relationships between categories. As long as they are different words, they are completely different under one hot coding [17]. For example, one might expect words like “table” and “chair” to have some similarity, but the one-hot encoding treats them as two words with 0 similarities. A distributed representation is used in neural language models. In the neural language model, each word is no longer a completely orthogonal 0-1 vector, but a point in a multi-dimensional real number space, which is embodied as a real number vector. In many cases, this distributed representation of words is also called word embedding. The distributed representation of words can be viewed as a point in Euclidean space, so the relationship between words can also be characterized by the geometric properties of the space. Different words can be represented on a 512-dimensional space. Under this representation, there is a certain connection between “table” and “chair.” The traditional machine learning method of natural language processing firstly trains a model for a specific language in a large number of parallel corpora and then applies the machine translation model to the translation task of the specific language. Compared with transfer learning, its basic conditions are no longer required. First, training materials and test data for machine learning standards should be distributed independently and independently; second, the balance in the body used for exercise must be measured and performed to achieve good results. The concept of transformational education allows for the use of existing data to train neural network models and transfer the learned experience to neural network models with less training corpus so that training materials can be reduced. And training time can be reduced. In general machine learning, for various positions, it is necessary to write various registration documents related to the training for attaining their independent standards. Compared to these ideas, learning changes can be a good model in the context of small data [18, 19]. Transfer learning stores the knowledge acquired by training model A and applies it to new tasks. The figure shows the training of model B to achieve the purpose of improving the performance of model B. The transfer learning strategy is very suitable for tasks that lack of existing labeled data. In addition to a small number of languages with rich parallel corpus data resources (such as Chinese, English, and German), the problem of lack of corpus resources in many languages is common, and there is not enough labeled data. The introduction of transfer learning will effectively alleviate this difficulty. Domain-specific machine translation systems are in high demand, while general-purpose machine translation systems have a limited range of applications.

TABLE 1: Comparison of inductive, direct push, and unsupervised transfer learning.

Transfer learning style	Source domain labeling	Labeling of target domain
Inductive transfer learning	Yes	Little
Direct push transfer learning	Yes	No
Unsupervised transfer learning	No	No

Generic systems are generally less performant and therefore important for domain-specific machine translation development [20]. Domain-specific adaptation is a key problem in machine translation. The goal is to study the specific domain of the model. As we all know, special reconstructive models (news, speech, medicine, literature, etc.) have more accuracy in neurological pathogens under the same name. Specifically, when the training data are distributed unbiasedly on the target domain, the final model will be compared against the test data during training on the dev set. Domain adaptation usually includes terminology, domain, and style adaptation. However, if the training data comes from a different source of purpose, the performance will be reduced accordingly. To build well-performing machine learning (ML) models, the model must be trained and tested against data from the same target distribution. For example, when the training data come from news articles and the test domain is specific to the medical domain, the translation performance will be unsatisfactory. We often have a large number of out-of-domain parallel statements. The challenge of training domain-specific models is to improve translation performance in the target domain given only a small amount of additional in-domain data. This can be accomplished by modifying the structure with special data (also called continuous training). Domain adaptation has been used successfully in computing and neural translation. In a typical neural machine translation domain adaptation setting, we first train the parent model on a resource-rich out-of-domain parallel corpus. On the basis of the general model, the training corpus is converted into an in-domain corpus and the parent model is fine-tuned. We can think of domain adaptation as transfer learning from an out-of-domain parent model to a domain-specific child model [21]. However, in real scenes such as online translation engines, the domain of sentences is not given. Guessing the domain of input sentences is very important for correct translation. In order to solve the problem of lack of data in the domain, the domain of a single sentence in the training data can be classified, and then the training sentences close to the target domain can be searched and selected. Inductive migration: the learning tasks of the source domain and the target domain are different but related. The labeled data of the target domain are available, but the labeled data of the source domain are not necessarily available. According to whether the label data of the source domain are available, it can be further divided into multitask learning (labeled data available) and self-learning (labeled data not available). Direct push transfer: when the target task and the source task are the same, the target domain data are unlabeled, but there is a large amount of available labeled data in the source domain. In this case, it is assumed that the tags of the same instance are the same across different

domains, meaning that the whole case of the same instance does not depend on the author. Unsupervised migration: the registry and destination functions are different but related, and there is no script, as shown in Table 1.

The main idea of instance-based transfer learning is to reduce the difference between the source domain and the target domain by changing the existing form of the samples, which is mainly suitable for situation where the similarity between the source domain and the target domain is high. The main idea of migration based on feature representation is to find a better feature representation, minimize the difference between domains and the error of classification and regression, and make the source domain and target domain show similar properties in a certain feature space through feature transformation, which can be applied to the case, where the similarity between domains is not too high or even dissimilar, it can be divided into supervised and unsupervised situations [22]. The transfer method based on model parameters assumes that the models on related tasks can share some parameters from the perspective of the model, so as to share some parameters between the source domain model and the target domain model to achieve the effect of transfer learning. The relationship-based transfer achieves the effect of transfer learning by establishing a map of the correlation knowledge between two domains. It does not assume that the data in each domain is independent and identically distributed, but transforms the relationship between the data from the source domain is migrated to the target domain, as shown in Table 2.

Isomorphic transfer learning: its source domain and target domain have the same feature space, that is, its feature dimension is the same, but its feature distribution is different, see Table 3 for details. The realization of isomorphic transfer learning needs to solve the problem of domain adaptive learning. Commonly used methods include example weighted domain adaptive learning, feature representation domain adaptive learning, parameter and feature decomposition domain adaptive learning, multisource domain adaptive learning, and heterogeneous learning. Transfer learning: the feature space, feature dimension, and feature distribution of the source and target domains are different. Therefore, realizing heterogeneous transfer learning needs to solve the problem of feature space alignment first, and then solve the problem of domain adaptive learning, which is more complicated than homogeneous transfer learning.

## 4. Experiment and Analysis

The NMT model represents a sentence as a long vector in a sentence, but the long vector does not represent the entire

TABLE 2: Comparison of four kinds of transfer learning based on examples, features, models, and relationships.

Transfer learning style	Characteristic
Case-based transfer learning	Give the source domain instance a certain weight and reuse it
Feature-based transfer learning	Reducing the gap between source domain and target domain based on feature transformation
Model-based transfer learning	Find out the shared parameters between the source domain and the target domain network model
Relationship-based transfer learning	Mining the relationship similarity between different fields

TABLE 3: Comparison between isomorphic transfer learning and heterogeneous transfer learning.

Transfer learning style	Source domain and target domain feature space	Problems to be solved
Isomorphic transfer learning	Identical	Domain adaptive learning
Heterogeneous transfer learning	Different	Adaptive learning of its + domain in feature space

semantic data of the sentence. The NMT-based monitoring process first encodes the sentence-by-sentence vector sequence, and then dynamically searches for relevant information related to word creation through the monitoring process while developing languages, which NMT's capabilities make much better. This document provides Chinese-English and English-Chinese translation standards required by Klein. First, two preschool courses (A and B) focused on large-scale Sino-English parallel corpora and more than one Anglo-Chinese parallel corpora; second, during the training of the Sino-English NMT model, the encoder parameters of the Sino-English translation standard are started by the encoder parameters of the Sino-English standard, and the decoder parameters of the Sino-English standard translation unit are started by the decoder. English-Chinese model endless is an excellent model to achieve the final TINMT\_CV model (C) starting with the Sino-English parallel corpus [23]. It can be assumed that the final result (BLEU value) of the Sino-English and Tibetan-Chinese neural translation using the transition model is better than the traditional translation without exchange of knowledge. Based on the extensive training and adaptation of the Sino-English model, BLEU's rate of improvement at an early stage was faster than the standard translation that relied on Sino-English and Tibetan-Chinese text for the neural network. Chinese-English resource training is used when using the BLEU value of 25 as the target for the joint corps. With standard modifications, standard translation can be completed in 20,000 steps, but without standard translation, standard translation can be learned in 80,000 steps. When the BLEU value reaches 40, the target point of Tibetan sugar is less when combining corpus training. However, with standard definition, interpretation can be completed in 50,000 steps, as shown in Figure 4.

Due to the influence of parameter initialization before machine translation model training, the parameters of large-scale Chinese-English translation model belonging to the same translation task are introduced into the initialization of low-resource Chinese-English and Tibetan-Chinese translation models so that the model has a certain parameter basis before training, so its learning rate will be improved during retraining [24, 25]. In this document, the encoder and decoder parameters of the Chinese-English translation model are initialized together with the parameters of the Chinese encoder of the Sino-English model and the decoder of the

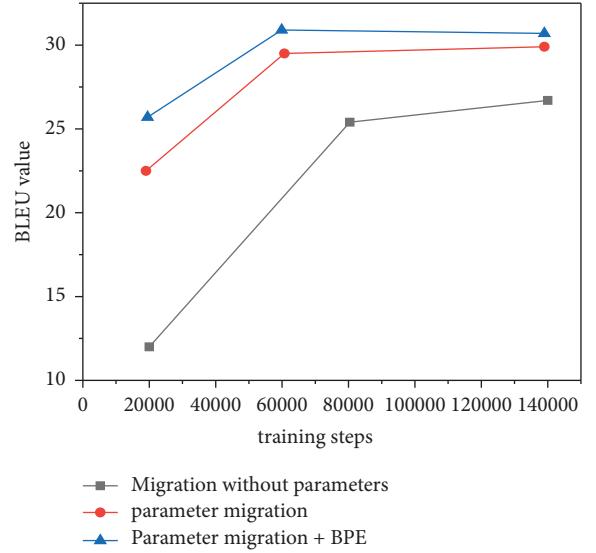


FIGURE 4: Relationship between low resource Chinese-English BLEU value and training steps.

English-Chinese model. As a basis for this, the small size of the Sino-English bilingual corpus for good training is used to achieve the Sino-English NMT standard. To improve the relationship between the encoder and the decoder received by the pretraining and to ensure that the initialization is better for good training, this article presents the training before testing. Firstly, the pivot language English is reinvigorated in the existing Chinese-English training set, and the large-scale English-Chinese parallel corpus is used to train the English-Chinese translation model; Then we use the English-Chinese translation model to retranslate the English in the English-Chinese parallel corpus, so as to obtain the Chinese-English-Chinese trilingual parallel corpus; then use the method of data enhancement 16 to increase the Chinese-English parallel corpus, improve the correlation between the model parameters, and reduce the existing noise. In this experiment, a Chinese English parallel corpus with a scale of 100,000 sentence pairs is used, of which 13,000 sentence pairs are tested and 11,000 sentence pairs are verified; 700,000 pairs of English-Chinese parallel corpora, including 5,000 pairs of test corpora and 4,000 pairs of verification corpora; There are 50 million pairs of Chinese-English parallel corpora, including 30,000 pairs of test

corpora and 10,000 pairs of verification corpora. Before the training, the experimental data are filtered for garbled code and word segmentation. In order to evaluate the effectiveness of the TINMT\_CV model, the experiment selects five baseline systems Moses, transformer, CNN, NMT trans, GNMT, and the TINMT\_CV model proposed in this paper. A total of 120,000 English-Chinese parallel corpora are used as training sets in the direction of English-Chinese translation. The terms used by the transformer, TINMT\_CV, and NMT trans model are set to 32000, the maximum number of lines is set to 50, “transformer\_ff” is set to 2048, “lab horizontal equalization” is set to 0.1, “led head” is set to 0.1, set to 2, “dropout” is set to 0.2, the number of layers is set to 2, the word embedded dimension is set to 256, “batch size” is set to 128, and the teaching value is set to 0.2. The optimizer selects Adam, with “NUM units” set to 128 and “dropout” set to 0.2. In this article, the two-dimensional high-efficiency test (BLEU) is used as a measurement tool. Table 1 shows the comparison results of the BIEU values between the baseline system and the TINMT\_CV model in both English-Chinese and Chinese-English translation directions. Among them, the TLNMTe is the TLNMT\_CV model, which is only pretrained encoder, and the TLNMTd is the TINMT\_CV model, which is only pretrained in the measurement encoder. It can be seen from the experiment that the results of the TLNMT\_CV model of the Anglo-Chinese bilingual NMT are better than the basic process, of the TLNMTe bi model. Compared to Moses’ example, the EU rate increased by 1.52% for English-Chinese translations and 1.31% for Chinese-English translations. Compared with the transformer model, the BLEU value of the TLNMTe model increased by 0.38 percentage points in the direction of the English-Chinese translation and 0.44 percentage points in the area of introduction of the Chinese-English translation. By quality, U-value for the TINMT\_CV model in the direction of English-Chinese translation is 0.71% higher content than the NMT trans model and 0.48% higher content in the introduction of Chinese-English translation. The TINMT\_CV model is used in the direction of English-Chinese translation. The EU rate increased by 1.16 percentage points compared to standard manpower and 1.05 percentage points in the direction of Chinese-English translation. This article presents Ms. TLNMT CV method, which can guide the first error in the Chinese-English NMT encoder and decoder using large-scale Chinese-English and English-Chinese corpora and can accept Chinese-English NMT standards through small-scale Chinese-English finetuning training. This method can improve the performance of low-resource Chinese-English NMT. Comparative experiments also proved the effectiveness of this project. In the next step, we can explore the widespread use of Chinese-English monolingual corpus for pretraining, and the knowledge gained from pretraining of the Sino-English bilingual NMT model to improve translation efficiency. Can be integrated into the construction. In this section, large Chinese-English corps are trained 200,000 steps to achieve stable standards, 100,000 steps are trained for rare Chinese-English and Tibetan-Chinese materials, and 5,000 for comparison experiments. The BLEU value of the steps was

TABLE 4: Chinese-English translation of low resources.

Model framework	BLEU
Transformer (not migrated)	27.39
Transformer (migration)	31.36
Transformer + BPE	31.68

TABLE 5: Low resource Tibetan-Chinese translation.

Model framework	BLEU
Transformer (not migrated)	46.02
Transformer (migration)	48.53
Transformer + BPE	48.92

recorded. Table 4 compares the benefits of educational change based on the function of the Chinese neurotransmitter. The Table shows the training test results under the English material resources in 10 W, as shown in Tables 4 and 5.

The comparison results of machine translation models are shown in the table mentioned above. It can be seen that the model transfer learning of low resource Chinese and English parallel corpora improves the translation of the nontransfer learning translation system by 3.97 BLEU values, and the translation of the translation system pretreated with BPE technology improves the translation of the translation system by 0.34 BLEU values compared with the translation system with only transfer learning. The model transfer learning of low resource Tibetan-Chinese parallel corpus improves the translation value of the nontransfer learning translation system by 2.64 BLEU values, and the neural machine translation system with BPE technology preprocessing and model transfer learning improves the translation value by 0.26 BLEU values compared with the translation system with only transfer learning. NMT is a typical encoding and decoding structure, in which the encoder reads the entire sentence sequence and encodes it to obtain the vector table of the sentence. The decoder uses the sentence vector obtained by the encoder as the target input and generates the words of the target language word by word. Sequence transfer learning can transfer the parameters learned by the model to similar tasks, and use the parameters obtained from high-resource translation tasks to improve the performance of low-resource translation tasks, thereby reducing the translation task’s dependence on parallel data, but fixed-length vectors cannot be used. Fully express the semantic information of the sentence in the source language. However, the semantic information of a sentence cannot be fully expressed in the source language using a fixed-length vector. The NMT-based monitoring process first encodes sentence by sentence into vector sequences, and then dynamically searches for contextual information related to word generation through the language development monitoring process, which greatly enhances the capabilities of NMT.

## 5. Conclusion

With the application of artificial intelligence and deep learning technology in more and more fields, machine

translation, as an important part of natural language processing, frequently appears in people's daily life applications, which is of great research value. At this stage, the mainstream machine translation methods have turned from traditional statistical methods to deep neural network methods. The main work is divided into the following parts: by reading Chinese and foreign literature related to machine translation, consulting reference materials, learning neural machine translation technology, fully understanding the main technologies proposed by academia and industry in the field of neural machine translation, and the application of these technologies, compare the proposed background, application scenarios, advantages and disadvantages of each model, learn various machine translation models according to the introduction of the references, and fully understand the multi-angle knowledge of machine translation. Through the research on various neural machine translation methods, it is found that when using pretraining deep learning technology to initialize the model, obtaining a high-quality pretraining model greatly affects the translation effect of the neural machine translation model, because pretraining is a pretrained and saved network that was previously trained on a large dataset, we can use the pretrained model as a feature extraction device for transfer learning. When the features learned by the pretraining model are easy to generalize, transfer learning can get better results. When using deep learning technology to deal with text translation problems, it is first necessary to convert the text into word vectors. The traditional recurrent neural network word vectors can only represent the frequency of occurrence of different words and the co-occurrence relationship between words, although the co-occurrence relationship is to a certain extent, it reflects the correlation between words, it still cannot accurately reflect the contextual relationship, which affects the accuracy of the algorithm for text translation. To solve this problem, this paper uses a model-based transfer method. First, the Chinese-English parallel with sufficient training data are used. The corpus task trains the transformer machine translation model, and then the model parameters are transferred to the model training of low-resource Chinese-English and Chinese parallel corpora. In this process, the idea of model transfer is used, that is, the parameters of the machine translation model trained on the massively parallel corpus are transferred to the training of the low-resource neural machine translation model, thereby improving the accuracy of the low-resource neural machine translation. On the other hand, the traditional recurrent neural network structures RNN, LSTM, and GRU have complex structures, many model parameters, cannot process data in parallel, and are difficult to train. Therefore, this paper uses the transformer model based on the attention mechanism for model training, which speeds up the training speed, and improves the translation effect. Then, experiments are used to demonstrate that the proposed low-resource neural machine translation method based on model transfer has higher

translation accuracy than the untransferred neural machine translation method.

## Data Availability

The data used to support the findings of this study are available from the author upon request.

## Conflicts of Interest

The author declares no conflicts of interest.

## Acknowledgments

This work was supported by the First-Class Course Foundation of Huanggang Normal University (no. 2020CK07).

## References

- [1] Q. Song, N. Zhang, and H. Liang, "Review of the Chinese internet philanthropy research (2006-2020): analysis based on citospace," *The China Nonprofit Review*, vol. 13, no. 1&2, p. 4, 2021.
- [2] X. Zhang, "Research on the language characteristics of agricultural English and its translation strategies," *Journal of Higher Education Research*, vol. 3, no. 1, pp. 5–8, 2022.
- [3] L. Forti, "Review of lu (2017): a corpus study of collocation in Chinese learner English," *International Journal of Learner Corpus Research*, vol. 8, no. 1, pp. 144–149, 2022.
- [4] L. Jin, "Research on pronunciation accuracy detection of English Chinese consecutive interpretation in English intelligent speech translation terminal," *International Journal of Speech Technology*, vol. 5, pp. 1–8, 2021.
- [5] R. Qing-dao-er-ji, Y. L. Su, and W. W. Liu, "Research on the lstm Mongolian and Chinese machine translation based on morpheme encoding," *Neural Computing & Applications*, vol. 32, no. 1, pp. 41–49, 2020.
- [6] D. Xia, "Research on Chinese-English translation compensation based on cognitive linguistics," *International Journal of Social Science and Education Research*, vol. 3, no. 4, pp. 235–242, 2020.
- [7] N. Ming, "Characteristics and trends of English testing research in China: visual analysis based on citospace," *Region - Educational Research and Reviews*, vol. 3, no. 3, pp. 1–6, 2021.
- [8] W. He and L. Chen, "A Research of Neural Style Transfer on Line Structure Based on Sequence to Sequence Learning," *IEEE Access*, vol. 8, no. 99, pp. 112309–112322, 2020.
- [9] B. Ji, Z. Zhang, X. Duan, M. Zhang, B. Chen, and W. Luo, "Cross-lingual pre-training based transfer for zero-shot neural machine translation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 115–122, 2020.
- [10] A. Kumar, A. Pratap, A. K. Singh, and S. Saha, "Addressing domain shift in neural machine translation via reinforcement learning," *Expert Systems with Applications*, vol. 201, Article ID 117039, 2022.
- [11] X. Wu, Y. Xia, J. Zhu, L. Wu, S. Xie, and T. Qin, "A study of BERT for context-aware neural machine translation," *Machine Learning*, vol. 111, no. 3, pp. 917–935, 2022.
- [12] K. Xu, "Needs analysis of Chinese English majors in egp courses," *Journal of Language Teaching and Research*, vol. 12, no. 3, pp. 452–465, 2021.

- [13] X. Qiao and R. Hao, "Move-step analyze of the introduction in Chinese and English accounting journals: based on analysis of accounting research and journal of accounting and economics (2016–2018)," *Open Journal of Modern Linguistics*, vol. 9, no. 6, pp. 540–550, 2019.
- [14] J. Luo and D. Li, "Universals in machine translation?:a corpus-based study of Chinese-English translations bywechat translate," *International Journal of Corpus Linguistics*, vol. 27, no. 1, pp. 31–58, 2022.
- [15] N. Peng, "Research on the effectiveness of English online learning based on neural network," *Neural Computing & Applications*, vol. 34, no. 4, pp. 2543–2554, 2021.
- [16] X. Zhao, "Research on the English translation of Chinese public signs based on face theory," *OALib*, vol. 8, pp. 1–7, 2021.
- [17] S. Huang, "A contrastive analysis of English novel the notebook and its Chinese translation from the perspective of rewriting theory," *Advances in Social Sciences Research Journal*, vol. 7, no. 12, pp. 320–332, 2020.
- [18] B. Zhang, "Analysis of web-based college English teaching from 2000 to 2017 in China," *English Education Journal of English Teaching and Research*, vol. 5, no. 1, pp. 1–12, 2020.
- [19] X. Geng, L. Wang, X. Wang et al., "Learning to refine source representations for neural machine translation," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 8, pp. 2199–2212, 2022.
- [20] F. Xu, X. Zhang, Z. Xin and Alan Yang, and A. Yang, "Investigation on the Chinese text sentiment analysis based on convolutional neural networks in deep learning," *Computers, Materials & Continua*, vol. 58, no. 3, pp. 697–709, 2019.
- [21] X. D. Li and H. H. Cao, "Research on vr-supported flipped classroom based on blended learning — a case study in learning English through news," *International Journal of Information and Education Technology*, vol. 10, no. 2, pp. 104–109, 2020.
- [22] A. Poncelas, G. M. D. B. Wenniger, and A. Way, "Improved feature decay algorithms for statistical machine translation," *Natural Language Engineering*, vol. 28, no. 1, pp. 71–91, 2022.
- [23] Y. Chang, "A research proposal on applying Chinese phonetic system in teaching pronunciation of English words to older Chinese efl adult learners," *Journal of Higher Education Research*, vol. 3, no. 1, pp. 21–25, 2022.
- [24] M. Ramos, "Teaching English for medical translation: a corpus- based approach," *Language Teaching Research Quarterly*, vol. 8, no. 2, pp. 25–40, 2020.
- [25] R. Baruah, R. K. Mundotiya, and A. K. Singh, "Low resource neural machine translation: Assamese to/from other indo-aryan (indic) languages," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 1, pp. 1–32, 2022.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/368873507>

# Ethical Issues for AI-Solutions in Business: Hype or...?

Conference Paper · February 2023

---

CITATIONS

0

READS

13

1 author:



Sergei Kladko

Innopolis University

12 PUBLICATIONS 0 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



AI-Human Cross-Cultural Interaction: Digital Expectations vs Analog Perception [View project](#)

# Ethical Issues for AI-Solutions in Business: Hype or..?

Sergei Kladko, Innopolis University (s2kladko@gmail.com)

## Abstract:

The abstract considers the issues of ethics in the solutions on the basis of artificial intelligence (AI) with regards to modern business environment. The author believes that the so-called ‘ethical’ challenges that find their practical implementation in AI solutions in various types of discrimination have already become a serious threat both for the IT-developers and for the companies that apply such tools in their everyday business practice. The recent years have witnessed an unprecedented growth in the number of cases when the ignorance to the ethical bias in AI solutions resulted in financial and reputational damage. To help business community overcome such challenges, the author, on the basis of the analysis of the corpus of practical, provides three suggestions regarding more active introduction of ethics in AI solutions and, on the other hand, avoiding the so-called ‘ethics washing’ as a potentially dangerous practice which impedes technological and business development.

**Keywords:** *AI ethics, artificial intelligence, technologies in business, business ethics*

## 1. AI Market: Lucrative Prospects and Expensive Biases

Today, the global market of artificial intelligence (AI), with a projected growth from \$432 billion in 2022 to \$900 billion by 2026 [International Data Corporation 2022], cannot but amaze with its prospects in the near future. At the same time in the Russian Federation the AI market volume is predicted to amount to 555.1 million dollars by 2024 [TAdviser 2022] (however, other forecasts look even more optimistic, assuming the market growth of up to 160 billion rubles [Ministry of Digital Development, 2019]). Correspondingly, the companies increase their expenditures on the implementation of AI solutions with the hope to gain a stable financial position in the future and, most importantly, remain competitive in their markets. For instance, according to the survey in the United States, more than a third of high-margin companies spend on AI from \$51 to 100 million, and seven out of ten organizations spend \$1 million or more of their budget on AI [Venturebeat 2022] (and this does not include smaller companies where spending on AI solutions, in most cases, exceed \$50 thousand per year).

There is no need to say that the abovementioned figures and prospects for the development of the AI solutions market are staggering. However, today various actors (gradually, but quite confidently) interacting within the AI ecumene are asking questions related not only to the financial prospects for the implementation of these solutions but also to the ways of how to make AI solutions more ‘responsible’ in terms of the moral and ethical attitudes of the relevant target audiences. One can, of course, argue that ethical issues in business have always been considered a kind of semantic load, which, under favorable circumstances, could also be monetized. However, the current practice demonstrates that underestimating (or completely ignoring) the ethical issues that various TA’s care about when interacting with AI solutions in education, medicine, litigation, or recruiting leads not only to reputational, but also to significant financial costs. This can be seen in the results of a joint survey by DataRobot and the World Economic Forum, in which the participants declare that the presence of biases in data has already cost them the loss of income (62% of respondents), customers (61%) or employees (43%), and some of them (35%) suffered serious financial costs caused by lawsuits [DataRobot 2022]. Moreover, 6% of respondents were forced to admit that underestimating data biases in AI solutions led to significant damage to the company’s brand, the biggest cost for the business today. On the other hand, behind the dry descriptions of cases of cultural and ethical failures of AI algorithms are dozens of real people who have lost their jobs, their good name, and sometimes their lives. The

resulting multimillion-dollar lawsuits and reputational losses, directly related to the inclusion of data filled with racial, national, gender and cultural biases in AI algorithms, clearly do not allow businesses to regard this problem as of secondary importance and not related to the financial stability of companies.

## **2. Key Challenges to Ethics in AI Solutions**

First of all, it is important to emphasize that, for this paper, AI ethical issues are considered in regards of the so-called narrow (weak) AI characterized by the limited use of data, depending on who created or processed it. Hence, one of the main problems in the development and further operation of an "ethically correct" AI solution is the significant subjectivism of the concepts of "morality". At the same time, as researchers today admit, it is impossible for any developer to get rid totally of their own cultural stereotypes when creating an appropriate AI solution, which, as a result, will unambiguously contain a set of cultural associations and prejudices of actors (developers primarily) [Caliskan, 2017, p. 184]. At the same time, the norms and values of the respective TA's will not always coincide with the cultural characteristics of the developers, and, in some cases, seriously contradict them. Accordingly, it is not possible to talk about the possibility of a unanimous opinion on the same concepts of "good" and "evil". It is not surprising, therefore, that all this causes a natural negative reaction both from developers who are being blamed for the "unethical" behavior of AI algorithms and from the business community, which is trying to find the right balance between community expectations and preventing the rise in the cost of innovative solutions. The global survey conducted by the Pew Research Center puts the concern about the generally accepted terminology of AI-ethics at the top of the list of the main problems in creating AI solutions that meet the ethical expectations of various actors. Other anxieties voiced by respondents also cast doubt on whether ethics may soon become an important issue when working on AI solutions. First of all, the main players in the market (large corporations and governments in general) are not very concerned with such issues themselves, preferring other criteria for evaluating the effectiveness and importance of the corresponding product or service. Finally, developers and businesses are absolutely right by stating that an excessive focus on resolving ethical issues in AI solutions will lead to a serious increase in the cost of the latter (even collecting the most diverse data means additional costs) and, as a result, to the final defeat in the technological race. It should be recognized that such fears are clearly not unfounded, and, as a result, the lack of clear answers to the questions above means that ethical issues continue to be sacrificed in the process of constant competition in the technological market. This is exactly the answer to the question of whether most AI systems will be based on ethical principles by 2030 by almost 68% of practitioners (including the heads of the largest IT companies) within the above question [Pew Research Center, 2021].

Finally, even the creation of an AI solution that could turn out to be as ethically correct as possible may not mean that its creators will not receive the same reproaches after some time as the developers who ignored the moral aspect in data processing. Today there are serious concern about the so-called 'ethics greenwashing' which can replace the process of searching for the balance between ethical expectations and the real-life situation, thus discrediting totally the very idea of AI-ethics. An illustration of this is Microsoft's attempt to fix the epic fail with the Tay chatbot by launching the Zo chatbot, an important characteristic of which was maximum neutrality in relation to the most sensitive topics for users (religion, race, etc.), while all concepts associated with individual cultures (from their history to positive judgments about them) fell under the ban. As a result, the new chatbot was accused of censorship, narrow-mindedness, and... insufficiently ethical behavior [Stuart-Ulin 2018 (2022)].

## **3. Case Study and Conclusions**

In an attempt to develop recommendations for a more balanced consideration of ethical factors in the development of AI solutions, the author analyzed a corpus of 24 practical cases collected from open sources over the period from 2017 to 2022. Cases were divided into groups in the following areas: medicine, law and litigation, education, and recruitment. The development of

each case was tracked in open sources (including lessons learned from it for other companies), on the basis of which a conclusion was made about what has changed for a particular area in terms of the "ethical" implementation of AI solutions over a five-year period. All this allowed the author to come to the following conclusions:

- 1) In the field of healthcare, the number of cases of unethical interpretation of data by AI algorithms remains quite high; however, testing has become more thorough in recent years (which, given the scope, is not surprising). Prejudices are connected, first of all, with the insufficient representativeness of the data in relation to the racial / national or gender affiliation of the person being tested. At the same time, medicine remains the only area where it is virtually impossible to find in the public domain the name of the project or the company that made such mistakes (there are only general description of cases).
- 2) Over the past three years, the number of cases of "unethical" use and interpretation of data in AI solutions in the field of education has increased significantly. In many ways, this was due to the COVID-19 pandemic, when, in order to save the educational process, it was sometimes necessary to put into practice not fully tested technologies. As a result, racial and social biases embedded in AI solutions have had a negative impact on the results of the most important assessment exams in the US [Reeves 2021] and the UK [Coughlan 2020], or, for example, when objectively considering a low GPA for university admission [Burke 2020]. Moreover, there was a case when an ethical bias in terms of income or place of study affected several thousand people in various countries (certification under the International Baccalaureate program in 2020 [Evgeniou 2020]). At the moment, the penetration of AI technologies in education continues to grow dynamically, which is likely to lead to the emergence of a considerable number of cases when the ethical component leads to significant conflicts.
- 3) Within the realm of litigation and law enforcement, after the COMPASS and HART scandals, discriminatory prospects based on the use of biased data seemed to be significantly reduced. However, a new scandal soon followed with the PredPol system (2018-2021), which continued the unethical activities of its predecessors, thus raising again the question of the correctness of using of such solutions in the field of law enforcement [Sankin 2021]. However, the developers of some new systems claim that they were able to take into account previous mistakes and that their systems have passed (and are passing) the most serious ethics test (although even in this case there are those who seriously question such claims) [Simonite T. 2019].
- 4) The field of recruitment continues to present a fairly serious ethical challenge for AI solutions. The history of Amazon's AI recruiting tools, accused of gender discrimination, has had little effect on the positive dynamics of the emergence of new AI solutions in this area. To date, applicants continue to complain about gender and age discrimination, as well as about ableism. However, the developers of such solutions tend to respond rather quickly to criticism regarding the ethical inconsistency of their products, which in fact strengthens the reputation of the brand.

#### **4. Solutions and Proposals**

In an attempt to resolve the issues about how AI solutions can best meet the ethical expectations of their respective target TA's, experts (both in business and in academia) offer various solutions. In relation to the distant future of AI products, it seems possible to talk about a certain convergence of human and machine behavior, which is already being recorded today by some scientists who note that a person surrounded by mechanisms imitates the behavior of electronic machines which, in their turn, acquire more and more human characteristics [Mazzara 2021]. Moreover, there is also the possibility that AI-based systems are beginning (albeit on a short-term basis) to influence the wider human culture [Brinkmann 2022], which, in principle, could resolve many of the current ethical discrepancies. However, all these transhumanist ideas do not solve the current problems that continue to negatively affect the development of AI solutions in industry and service.

First of all, it seems necessary to assist businesses and developers in developing methodological solutions for the formation of datasets with diverse and maximally free from ethical prejudices data. To date, the author is a member of a group that develops such methods by order of enterprises. These instructions contain not only recommendations on creating such datasets, but also a selection of practical cases, on the basis of which business customers can form a clear understanding of what a specific bias in a data set means and, most importantly, what the consequences of ignoring this problem.

Further, it can be interesting to develop the methods for a preliminary cultural and ethical audit of AI solutions. As part of these decisions, the author is participating in the working out of the foundations for the so-called cultural and ethical "red teaming". It should be noted that the classic red-teaming technique itself has long been very successfully used to find weaknesses in IT solutions, and it can be assumed that it can be reworked in order to identify weaknesses in forecasts and assumptions regarding the correct interpretation of the cultural and ethical intentions of customers and users. The main task of such teams (either formed within the company or a specially trained and specially certified third party) should be a simulated scheduled verification of an AI solution from the point of view of its most adequate subsequent normative-value perception by the relevant target audience.

Finally, it is important to intensify the creation of an institution of specialists in cultural and ethical facilitation in the process of implementing AI solutions. The presence of a trained certified expert mediator, who can help actors detect cultural and ethical contradictions when implementing a project and not allow ethical consideration transform into 'ethics washing', allows developers and customers not only to receive professional assistance, but also, most importantly, to significantly reduce the time and financial costs of independently conducting ongoing ethical AI audits for the solutions.

## References

1. Brinkmann L., Gezerli D., Müller K., Rahwan I., Pescetelli N. 2022 Hybrid social learning in human-algorithm cultural transmission // Royal Society Publishing. May 25. <https://doi.org/10.1098/rsta.2020.0426> (accessed September 12, 2022)
2. Burke L. The Death and Life of an Admissions Algorithm // Inside Higher Ed, December 14, 2020 <https://www.insidehighered.com/admissions/article/2020/12/14/u-texas-will-stop-using-controversial-algorithm-evaluate-phd> (Accessed November 07, 2021)
3. Caliskan, A, Bryson J., Narayanan A. Semantics derived automatically from language corpora contain human-like biases// Science, 356 (6334). 2017. C. 183-186.
4. Coughlan S. Why did the A-level algorithm say no? // BBC August 14, 2020. <https://www.bbc.com/news/education-53787203> (Accessed November 11, 2021).
5. DataRobot's State of AI Bias Report Reveals 81% of Technology Leaders Want Government Regulation of AI Bias //DataRobot January 18, 2022 <https://www.datarobot.com/newsroom/press/datarobots-state-of-ai-bias-report-reveals-81-of-technology-leaders-want-government-regulation-of-ai-bias/> (accessed September 03, 2022)
6. Evgeniou T., Hardoo D., Ovchinnikov A. What Happens When AI is Used to Set Grades? //Harvard Business Review August 13, 2020 <https://hbr.org/2020/08/what-happens-when-ai-is-used-to-set-grades> (Accesses September 13, 2022).
7. International Data Corporation. 2022. IDC Forecasts 18.6% Compound Annual Growth for the Artificial Intelligence Market in 2022-2026 Report, July 29 <https://www.idc.com/getdoc.jsp?containerId=prEUR249536522> (accessed August 30, 2022)
8. Mazzara M, Zhdanov P, Bahrami M.R. , Aslam H, Kotorov Iu, Imam M, and Salem H. Education after COVID-19 / In book: Smart and Sustainable Technology for Resilient Cities and Communities. Springer. June 2021. [https://www.researchgate.net/publication/351587748\\_Education\\_after\\_COVID-19](https://www.researchgate.net/publication/351587748_Education_after_COVID-19) (accessed August 20, 2022)

9. Ministry of Digital Development, Communications and Mass Media of the Russian Federation. 2019. Roadmap for the Development of the Digital Technology “Neurotechnologies and Artificial Intelligence” (in Russian), p.10 <https://digital.gov.ru/ru/documents/6658/> (accessed August 30, 2022)
10. Pew Research Center. 2021. Experts Doubt Ethical AI Design Will Be Broadly Adopted as the Norm Within the Next Decade (Report). June 16. <https://www.pewresearch.org/internet/2021/06/16/experts-doubt-ethical-ai-design-will-be-broadly-adopted-as-the-norm-within-the-next-decade/> (accessed August 01, 2022)
11. Sankin A., Mehrota D., Mattu S., Cameron D., Gilbertson A., Lempres D, Lash J. 2021. Crime Prediction Software Promised to Be Free of Biases. New Data Shows It Perpetuates Them. Gismodo, February 12. <https://gizmodo.com/crime-prediction-software-promised-to-be-free-of-biases-1848138977> (accessed September 02, 2022)
12. Simonite T. 2019. The Best Algorithms Struggle to Recognize Black Faces Equally. Wired. July 22. <https://www.wired.com/story/best-algorithms-struggle-recognize-black-faces-equally/> (accessed August 31, 2022)
13. Stuart-Ulin Chloe. 2018 (updated 2022). Microsoft’s politically correct chatbot is even worse than its racist one. Quartz, July 21 (July 31). <https://qz.com/1340990/microsofts-politically-correct-chat-bot-is-even-worse-than-its-racist-one/> (accessed August 13, 2022)
14. Richard V. Reeves and Dimitrios Halikias Race gaps in SAT scores highlight inequality and hinder upward mobility <https://www.brookings.edu/research/race-gaps-in-sat-scores-highlight-inequality-and-hinder-upward-mobility/> (Accessed September 10, 2022).
15. TAdviser. 2022. Solutions on the Basis of Artificial Intelligence (in Russian). January 08 [shorturl.at/ikuw9](http://shorturl.at/ikuw9) (accessed September 04, 2022)
16. Venturebeat. 2022. Report: 70% of orgs are spending \$1M or more on AI. February 22. <https://venturebeat.com/ai/report-70-of-orgs-are-spending-1m-or-more-on-ai/> (accessed September 06, 2022)

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/352665221>

# Fine Tuning Modeling Through Open AI

Chapter · June 2021

---

CITATIONS

0

READS

401

2 authors:



Varsha Desai

V.P.Institute of Management Studies & Research, Sangli

18 PUBLICATIONS 13 CITATIONS

[SEE PROFILE](#)



Kavita Oza

Shivaji University, Kolhapur

99 PUBLICATIONS 175 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Big Data : A text mining approach [View project](#)



Fine Tuning Modeling Through Open Ai [View project](#)

---

## Fine Tuning Modeling Through Open AI

---

Varsha P. Desai

Research Scholar, Computer  
Science Dept. Shivaji  
University, Kolhapur

Dr. Kavita S. Oza

Assistant Professor,  
Computer Science Dept.  
Shivaji University, Kolhapur

---

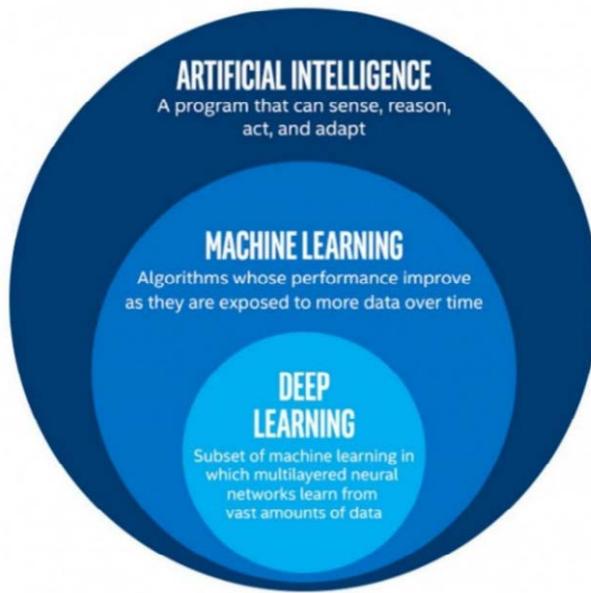
**ABSTRACT** - Open AI is an artificial intelligence research laboratory working on cutting-edge deep learning techniques that allows computer to work like a human being and help to solve complex problems. Natural Language Processing is one of the applications of deep learning through Open AI that motivates to train billions of parameters with complex and large corpus dataset with better performance. GPT-3 is auto regressive language has capability to convert text to image, face recognition, handwriting recognition, translation, sentence analysis, intelligent recommender. Responsive AI system uses behavior analytics for faster corporate decisions. This paper elaborates the fine-tuning applications of deep learning model through GPT-3 under open AI system.

Keywords: GPT-3, NLP, Deep Learning, ROUGE, Responsive AI

### INTRODUCTION

Today deep learning and neural network are the power of industries. Open AI is an advanced more human centric intelligent technique especially in field of reinforcement learning. It is artificial intelligence laboratory where research scientists explore their knowledge and skill for innovations in machine learning techniques. Artificial intelligence is the technique to allow the computer to behave like human being. Deep learning algorithms works like human brain that analyze complex data with huge logical combinations. AI has competency to enable deep learning model to solve complex mathematical problems better than machine learning. Deep learning models integrated with automatic optimization of feature extraction process rather than machine learning model.<sup>[1]</sup> Multilayered Neural network implementation using deep learning algorithms promotes more accuracy in prediction of complex problem results. Open AI technology motivates for creating image from text, connecting image to text, text analysis, language recognition, microscope etc.

Artificial intelligence is the simulation of human intelligence into machine for thinking and working like human that has capability to work from experience. Machine Learning is a part of AI that has an ability to learn from data, identify pattern and take decision with minimum human intervention. Deep learning is the subset of machine learning in which multilayer neural network learn from large corpus dataset for making intelligent decisions.

Fig.1 Cousins of AI <sup>[1]</sup>

### GPT-3

Generative Pre-trained Transformer 3 (GPT-3) is an autoregressive language model to produce human like text using deep-- learning algorithms. GPT-n series created by Open AI in the research laboratory. It is one of the effective technologies for Natural Language Processing(NLP).Autoregressive model has a capability to predict the outcome by random processing. Today it is a challenge to work on many NLP data and training the large corpus dataset by fine tuning on particular task to produce results. GPT-3 is a deep learning model with 175billion parameters, more than 10 thousands non sparse language model with fast and efficient test performance.<sup>[2]</sup> It provide strong performance on many NLP datasets like translation, comments, forums, cloze activity tasks, domain adaption and reasoning data analysis.<sup>[3]</sup>

Integration of Figma plugin and GPT-3 used to create interactive templates. It automatically generates code as well as provide comprehension of code written by programmer in python. It has capability to generate (JavaScriptXML) JSX layout from plain English text. Using GPT-3 we can generate regular expressions from different use cases written in plain English sentences. By combining capabilities of Figma and GPT-3 use to generate clone website from existing URL. It is intelligent techniques which tell us what things can be done with the inputted object. GPT-3 has a power to generate Automatic chart and plots from plain English. It can be used in interactive quiz designing for developing personalized eLearning applications. Learn from anyone is the GPT-3 tools where learner can select expert in particular area and to get knowledge from them by just typing query in plain English. GPT-3 is can write ML model for specific dataset and has capability to generate code for ML model only through dataset explanation and required output. It work as intelligent recommender,

Interactive Voice Response (IVR) provider, automatic resume creator without training and massive data uploads with greater accuracy<sup>[4]</sup>

## APPLICATIONS - OPEN AI

Jukebox model is developed using deep learning technique to generate raw audio in different styles and artists. Hierarchical Vector Quantised-Variational Auto Encoder (VQ-VAE) algorithms compress the music into tokens. This model generates the song piece of multiple minutes long which can recognizable singing in natural language voices.<sup>[5]</sup>

Reinforcement learning(RL) used to handle complex task by human judgment to provide positive and negative comments to the task, summarization of task. Human labels are used to train the models of reward and optimize the model. Deep learning algorithms are implemented to learn from human interaction. RL fine tuning of language model is developed for NLP tasks like high sentiment analysis, CNN/Mail summarization, TensorFlow (TL;DR) dataset. Both supervised learning and reward based learning algorithms provide better results for the NLP task. Communication between human and ML Model gives scalable reward learning methods such as amplification, debate and recursive reward modeling.<sup>[6]</sup>

GPT-3 is an autoregressive language model has an ability to work on large corpus dataset with billions of parameters with better test performance. It achieves strong performance on large number of NLP datasets for translation, cloze task, question-answering that required on-fly reasoning or domain adaption like 3-digit arithmetic, finding novel words in the sentence, unscrambling words. In Fine-Tuning approach supervised dataset is trained by updating the weight of pre-trained model until it results better performance. In few shot approach few demonstration of the task with inference time condition without updating weights. In one-shot approach model assign a single task at a time without updating gradient.<sup>[7]</sup>

Responsive AI system development ensures that system provides benefits to the society without any harm or negative impact. It involved testing the safety and security of system during development by identifying structural risk associated with AI system. AI security is very important to protect from being attacked, misused by bad actors and co-opted. In real time AI application sometime it is difficult to internalize harm from AI system like social harm due to increased use of AI system may reduce trust in online sources. AI targeted regulations includes government regulations, international standards and clarity on applying existing law to AI system. Responsive AI system development is an integration of action problem that expect benefit from being equilibrium. AI companies are facing many actions problems like more confidence that other will cooperate, assign higher value to the mutual cooperation, low expected cost, assigning low expected value to non-reciprocating cooperation, assigning lower expected value to the mutual defection.<sup>[8]</sup>

**Human Feedback Summarization:** Trained the model to prefer human feedback summary by using reinforcement learning. Extensive analysis use to understand human feedback dataset and fine-tuned model. Recall Oriented Understudy for Gusting

Evaluation (ROUGE) is the technique used for automatic summarization of text and machine translation of human produced text. Our reward model is developed that result in better summaries from human dataset. Reddit-trained human feedback models also generate high-quality summaries of news articles on the CNN/Daily Mail (CNN/DM) dataset without any news-specific fine-tuning, almost matching the quality of the dataset's reference summaries. Optimizing our reward model directly results in better summaries than optimizing ROUGE according to humans.<sup>[9]</sup>

Due to industrial revolution Artificial intelligence demand is increasing day by day. These include financial services, defense, consumer retail, advertising, entertainment. According to market research report it is forecasted that global revenue of artificial intelligent product and services will grow to \$36.8 billion by 2025.<sup>[10]</sup>

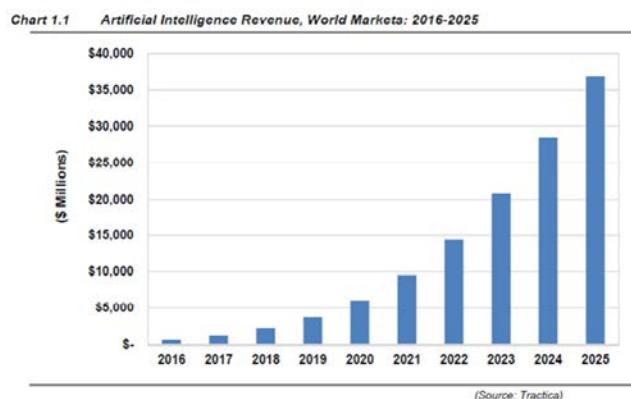


Fig.2 AI Revenue Growth World Market 2016-25<sup>[10]</sup>

According to market survey repot of Tractica Artificial Intelligence revenue for top 10 use cases of AI by 2025 generated form applications like contract analysis, Object detection and classification, Automated geophysical feature detection, text query of images, content distribution on social media, predictive maintenance, processing patient data ,image recognition, classification, tagging, algorithm treading strategy performance improvement.<sup>[10]</sup>

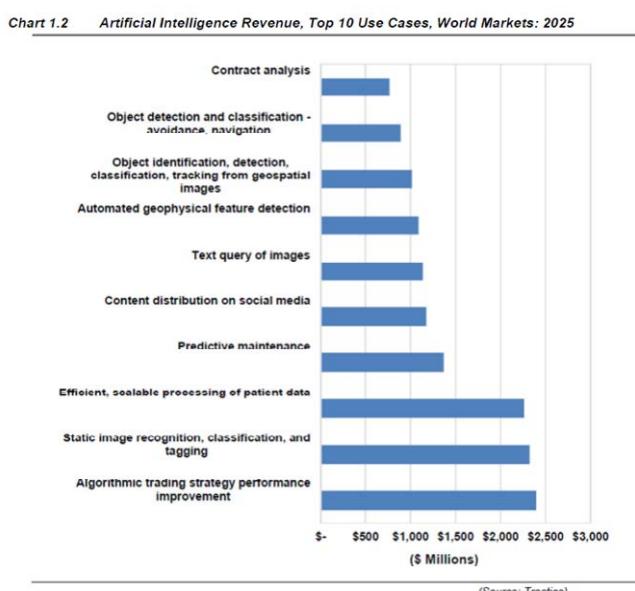


Fig.3 AI Revenue Top 10 Use Cases, World Market 2025<sup>[10]</sup>

## CONCLUSION

GPT-3 is an autoregressive model that works on large corpus NLP datasets with strong performance. In Reinforcement learning human labels are used to train the models of reward and optimize model to provide feedback. Open AI system motivates many fine tune modeling applications like face recognition, text summarization, cloze activity task, sentence analysis. ROUGE is used for automatic summarization of text and to translate human text into machine readable language. There is huge scope for deep learning algorithms and GPT-3 to work on large corpus data for object detection, speech recognition, language translation and complex decisions with better result. Due to industrial revolution, smart phone technologies, IOT, cloud based applications there will be huge investment in artificial intelligent projects by 2025.

## REFERENCES

- 1) <https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac>
- 2) <https://github.com/openai/gpt-3>
- 3) [https://en.wikipedia.org/wiki/Autoregressive\\_model](https://en.wikipedia.org/wiki/Autoregressive_model)
- 4) <https://www.educative.io/blog/top-uses-gpt-3-deep-learning>
- 5) Prafullahariwal, Heewoo Jun et.al.(Apr 2020), “Jukebox: A Generative Model for Music”,[Online] Available: <https://openai.com/papers/>, arXiv:2005.00341
- 6) Daniel M. Ziegler, Nisan Stiennon,et.al. (Jan 2020),“Fine-Tuning Language Models from Human Preferences”, [Online] Available: <https://openai.com/papers/>, arXiv: 1909.08593v2 [cs.CL]
- 7) Tom B. Brown, Benjamin Mann, et.al. (July 2020)” language Models are Few-Shot Learners”, [Online] Available:<https://openai.com/papers/>, arXiv: 2005.14165v4 [cs.CL]
- 8) Amanda Askell, Miles Brundage etal.,(July 2019) “The Role of Cooperation in Responsible AI Development” Source: <https://openai.com/papers/>, arXiv:1907.04534v1 [cs.CY]
- 9) Nisan Stiennon, Long Ouyang,et.al., (Oct 2020)“ Learning to summarize from human feedback”,Source:<https://openai.com/paper>, arXiv:2009.01325v2 [cs.CL]
- 10) <https://www.top500.org/news/market-for-artificial-intelligence-projected-to-hit-36-billion-by-2025/>

XXXXXXXXXXXX

# International Journal of Population Data Science

Journal Website: [www.ijpds.org](http://www.ijpds.org)



## GRAIMatter: Guidelines and Resources for AI Model Access from TrusTED Research environments (GRAIMatter).

Emily Jefferson<sup>1</sup>, Christian Cole<sup>1</sup>, Alba Crespi Boixader<sup>1</sup>, Simon Rogers<sup>2</sup>, Maeve Malone<sup>1</sup>, Felix Ritchie<sup>3</sup>, Jim Smith<sup>3</sup>, Francesco Tava<sup>3</sup>, Angela Daly<sup>1</sup>, Jillian Beggs<sup>4</sup>, and Antony Chuter<sup>4</sup>

<sup>1</sup>University of Dundee

<sup>2</sup>NHS Scotland

<sup>3</sup>University of West of England

<sup>4</sup>PPIE Co-I

## Objectives

To assess a range of tools and methods to support Trusted Research Environments (TREs) to assess output from AI methods for potentially identifiable information, investigate the legal and ethical implications and controls, and produce a set of guidelines and recommendations to support all TREs with export controls of AI algorithms.

## Approach

TREs provide secure facilities to analyse confidential personal data, with staff checking outputs for disclosure risk before publication. Artificial intelligence (AI) has high potential to improve the linking and analysis of population data, and TREs are well suited to supporting AI modelling. However, TRE governance focuses on classical statistical data analysis. The size and complexity of AI models presents significant challenges for the disclosure-checking process. Models may be susceptible to external hacking: complicated methods to reverse engineer the learning process to find out about the data used for training, with more potential to lead to re-identification than conventional statistical methods.

## Results

GRAIMatter is:

- Quantitatively assessing the risk of disclosure from different AI models exploring different models, hyperparameter settings and training algorithms over common data types

- Evaluating a range of tools to determine effectiveness for disclosure control
- Assessing the legal and ethical implications of TREs supporting AI development and identifying aspects of existing legal and regulatory frameworks requiring reform.
- Running 4 PPIE workshops to understand their priorities and beliefs around safeguarding and securing data
- Developing a set of recommendations including
  - suggested open-source toolsets for TREs to use to measure and reduce disclosure risk
  - descriptions of the technical and legal controls and policies TREs should implement across the 5 Safes to support AI algorithm disclosure control
  - training implications for both TRE staff and how they validate researchers

## Conclusions

GRAIMatter is developing a set of usable recommendations for TREs to use to guard against the additional risks when disclosing trained AI models from TREs.

PAPER • OPEN ACCESS

## Hyperparameter optimization of data-driven AI models on HPC systems

To cite this article: Eric Wulff *et al* 2023 *J. Phys.: Conf. Ser.* **2438** 012092

View the [article online](#) for updates and enhancements.

You may also like

- [Training-free hyperparameter optimization of neural networks for electronic structures in matter](#)

Lenz Fiedler, Nils Hoffmann, Parvez Mohammed et al.

- [Efficient hyperparameter tuning for kernel ridge regression with Bayesian optimization](#)

Annika Stuke, Patrick Rinke and Milica Todorovi

- [Random Forests Applied to High-precision Photometry Analysis with Spitzer IRAC](#)

Jessica E. Krick, Jonathan Fraine, Jim Ingalls et al.

## Breath Biopsy® OMNI®

The most advanced, complete solution for global breath biomarker analysis

BREATH BIOPSY®

TRANSFORM YOUR RESEARCH WORKFLOW



Expert Study Design & Management



Robust Breath Collection



Reliable Sample Processing & Analysis



In-depth Data Analysis



Specialist Data Interpretation

# Hyperparameter optimization of data-driven AI models on HPC systems

**Eric Wulff<sup>1</sup>, Maria Girone<sup>1</sup> and Joosep Pata<sup>2</sup>**

<sup>1</sup>CERN, Esplanade des Particules 1, 1211 Geneva 23, Switzerland

<sup>2</sup>NICPB, Rävala pst 10, 10143 Tallinn, Estonia

E-mail: [eric.wulff@cern.ch](mailto:eric.wulff@cern.ch)

**Abstract.** In the European Center of Excellence in Exascale Computing "Research on AI- and Simulation-Based Engineering at Exascale" (CoE RAISE), researchers develop novel, scalable AI technologies towards Exascale. This work exercises High Performance Computing resources to perform large-scale hyperparameter optimization using distributed training on multiple compute nodes. This is part of RAISE's work on data-driven use cases which leverages AI- and HPC cross-methods developed within the project. In response to the demand for parallelizable and resource efficient hyperparameter optimization methods, advanced hyperparameter search algorithms are benchmarked and compared. The evaluated algorithms, including Random Search, Hyperband and ASHA, are tested and compared in terms of both accuracy and accuracy per compute resources spent. As an example use case, a graph neural network model known as MLPF, developed for Machine Learned Particle-Flow reconstruction, acts as the base model for optimization. Results show that hyperparameter optimization significantly increased the performance of MLPF and that this would not have been possible without access to large-scale High Performance Computing resources. It is also shown that, in the case of MLPF, the ASHA algorithm in combination with Bayesian optimization gives the largest performance increase per compute resources spent out of the investigated algorithms.

## 1. Introduction

One of the primary goals in the European Center of Excellence in Exascale Computing "Research on AI- and Simulation-Based Engineering at Exascale" (CoE RAISE) [1] is the development and expansion of Artificial Intelligence (AI) and High-Performance Computing (HPC) methods along representative use cases from research and industry. While Work Package 3 (WP3) "Compute-Driven Use-Cases at Exascale" covers use cases that are compute-driven, WP4 "Data-Driven Use-Cases at Exascale" has a strong focus on data-driven technologies, i.e., analyzing data-rich descriptions of physical phenomena. Example use cases vary widely and range from fundamental physics and remote sensing to 3D printing and acoustics.

The work of WP4 is highly integrated with WP2 "AI- and HPC-Cross Methods at Exascale". Experts in WP2 provide support on HPC and AI methods to use cases in WP4. This support manifests itself in porting code to new HPC architectures and machines, in performance analyses and engineering of codes, and in the development of AI solutions for the individual use cases.

In the work presented here, HPC resources are leveraged to perform large-scale hyperparameter optimization (HPO) using distributed training on multiple nodes as part of WP4. As an example use case from the field of High Energy Physics (HEP), the AI-based



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

particle-flow reconstruction algorithm called Machine-Learned Particle-Flow (MLPF) [2] acts as the base model for which HPO is performed. MLPF is developed in the Compact-Muon-Solenoid (CMS) Collaboration [3] at CERN and combines information from tracks and calorimeter clusters to reconstruct particle candidates.

Further developments of AI in RAISE have the potential to greatly impact the field of High Energy Physics by efficiently processing the large amounts of data that will be produced by particle detectors in the coming decades. Moreover, HPO is model agnostic and could be widely applied in other sciences using AI, e.g., in the fields of seismic imaging, remote sensing, defect-free additive manufacturing and sound engineering that are part of Work Package 4.

HPO, sometimes referred to as hyperparameter tuning or hypertuning, is the process of tuning the hyperparameters of the model to optimize its performance. Hyperparameters are the parameters that are not learned during the model training but must be defined by the user. Some examples are model-architecture-related parameters such as the number of layers in the model and the number of nodes in each layer, or optimization-related parameters such as the batch size and the learning rate.

Hypertuning deep learning-based AI models is often compute resource intensive, partly due to the high cost of training a single hyperparameter configuration to completion and partly because of the infinite set of possible hyperparameter combinations to evaluate. There is therefore a need for large-scale, parallelizable and resource efficient hyperparameter search algorithms.

This work makes use of a distributed computing tool called Ray [4], and more specifically the part of Ray called Tune [5]. Tune is an open-source tool for multi-node distributed hypertuning which integrates well with modern machine learning frameworks like e.g., TensorFlow [6] and PyTorch [7]. It also supports integration with many other hypertuning tools such as Scikit-Optimize [8], HyperOpt [9], Optuna [10], SigOpt [11], and more.

In the following, section 2 describes the example use case for which HPO is performed, section 3 describes how HPO can be used in a wide variety of applications and highlight synergies within CoE RAISE, and finally, section 4 presents the conclusions.

## 2. Example use case: Event reconstruction and classification at the CERN HL-LHC

With the upcoming upgrade of the Large Hadron Collider (LHC) to the High Luminosity LHC (HL-LHC), the HEP community will face a significant increase in data production. This motivates efforts to optimize the speed and efficiency with which data is collected, processed, and analyzed, and is one of the major challenges that must be solved by the time the HL-LHC starts operation at the end of 2027.

One of the many different approaches that are being investigated to tackle this challenge is to replace traditional HEP algorithms with faster, parallelizable AI-driven approaches. These approaches promise to deliver similar or even better physics performance and can relatively easily be accelerated by hardware such as Graphics Processing Units (GPUs) or Field Programmable Gate Arrays (FPGAs).

One such traditional algorithm that could potentially be replaced by an AI-based version is the so-called Particle-Flow (PF) reconstruction algorithm [12]. It processes signals from different sub-detectors and combines them to construct higher-level physics objects. These objects are used for downstream workflows and are important for physics analyses involving hadronic jets and missing transverse energy. An effort to construct a machine-learned PF algorithm is the so-called MLPF algorithm, which is based on a deep neural network implemented using a Graph Neural Network (GNN) formalism. A detailed description of its first iteration can be found in [2] while a more recent version is described in [13]. The code to build, train, and evaluate the model is publicly available online [14].

The best performing MLPF hyperparameters were found after two stages of hypertuning.

The first stage was performed on the Jülich Wizard for European Leadership Science (JUWELS) Booster [15] at the Jülich Supercomputer Centre in Jülich, Germany, and required 19,574 core-hours to complete. Each compute node on the JUWELS Booster has two AMD EPYC Rome 7402 CPUs with 48 cores clocked at 2.8 GHz and four NVIDIA A100-SXM4-40GB GPUs. The so-called Bayesian Optimization Hyperband (BOHB) [16] algorithm was used to tune parameters of the optimizer such as the `lr` and the learning rate schedule as well as the `dropout` and other model-specific internal hyperparameters. The BOHB search space is summarized in table 1.

The second hypertuning stage was performed on CoreSite at the Flatiron Institute in New York, NY, USA, using twelve compute nodes, each equipped with a 64-core Intel IceLake Platinum 8358 CPU clocked at 2.6 GHz and four NVIDIA A100-SXM4-40GB GPUs. The best hyperparameter values found from the first search were fixed and stage two instead tuned various architecture parameters such as the number of graph layers and the number of graphs in each layer, as well as the number of nodes and layers used for decoding, and a few other model-specific parameters. The search space of the second stage is summarized in table 2. In addition, a different hypertuning algorithm called Asynchronous Successive Halving Algorithm (ASHA) [17] was used in combination with Bayesian optimization. The ASHA algorithm allows for an efficient use of compute resources when performing distributed multi-node hypertuning by early stopping trials that underperform relative to others. The second stage of hypertuning consumed approximately 56,730 core-hours. This work would not have been possible without access to HPC resources, as can be illustrated by a back-of-the-envelope calculation to compute that the two hypertuning stages would have taken roughly 6 months to complete using a single GPU, compared to about 83 hours using supercomputers.

Table 1: Search space used in the hypertuning run using the BOHB algorithm.

Hyperparameter	Search space
<code>lr</code>	$\log lr \sim U(10^{-4}, 3 \cdot 10^{-2})$
<code>dropout</code>	(0, 0.5)
<code>clip_value_low</code>	(0, 0.2)
<code>dist_mult</code>	(0.01, 0.2)

Table 2: Search space used for ASHA in combination with Bayesian optimization.

Hyperparameter	Search space
<code>bin_size</code>	{16, 32, 40, 64, 80}
<code>distance_dim</code>	{32, 64, 128, 256}
<code>ffn_dist_hidden_dim</code>	{32, 64, 128, 256}
<code>ffn_dist_num_layers</code>	{1, 2, 3, 4}
<code>num_graph_layers_common</code>	{1, 2, 3, 4}
<code>num_graph_layers_energy</code>	{1, 2, 3, 4}
<code>num_node_messages</code>	{1, 2, 3, 4}
<code>output_dim</code>	{32, 64, 128, 256}

In both stages described above, the search algorithms were allowed to draw 200 samples from the hyperparameter search space. The best hyperparameters found according to validation loss after both stages of hypertuning are reported in table 3 and various metrics as a function of the training epoch are shown in figure 1.

To see if HPO improved the model performance, the loss and classification accuracy of the model before and after hypertuning is plotted and compared as a function of the training epoch in figure 2. Comparing these curves shows that the mean validation loss decreased by almost a factor of two (approximately 44%) and that the accuracy increased by more than the uncertainty. It is also clear from comparison of figures 2a and 2b as well as of figures 2c and 2d that the training became more stable as a result of hypertuning since the curves exhibit much less volatility after hypertuning, especially in the second half of the training.

### 3. Distributed training and hypertuning: synergies across sciences in RAISE

HPO algorithms are model-agnostic in their nature and could be applied in any field of science making use of AI. Hence, the benchmarking of HPO algorithms is of interest for all use cases

Table 3: Best hyperparameters found.

Hyperparameter	Value
lr	0.001129
dropout	0.016312
clip_value_low	0.001998
dist_mult	0.120898
bin_size	64
distance_dim	64
ffn_dist_hidden_dim	128
ffn_dist_num_layers	3
num_graph_layers_common	3
num_graph_layers_energy	2
num_node_messages	3
output_dim	64

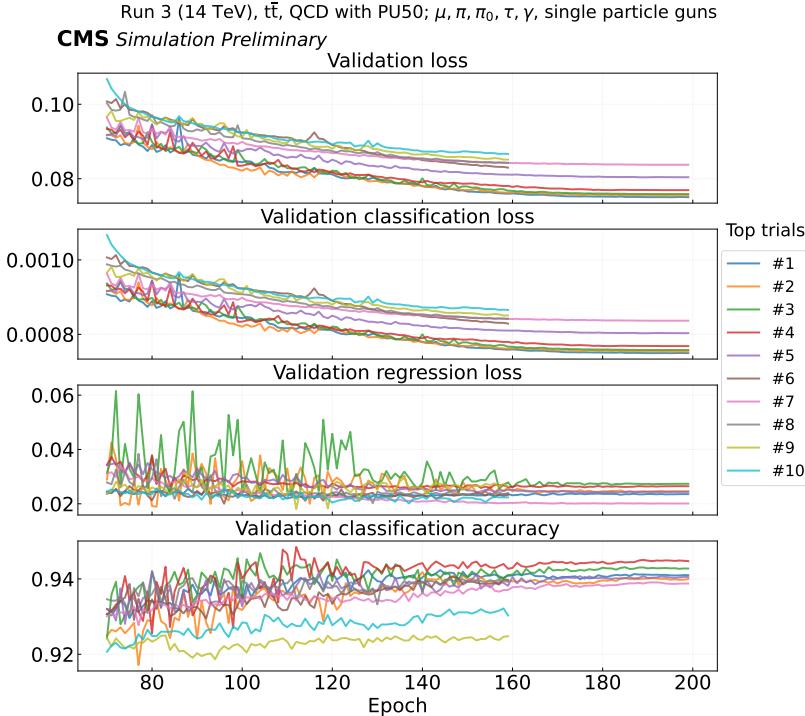


Figure 1: Loss and accuracy curves for top performing trials according to validation loss after hypertuning using the ASHA algorithm in combination with Bayesian optimization drawing 200 samples from the search space. From top to bottom: validation loss, validation classification loss, validation regression loss and validation classification weighted accuracy. The trials were trained for up to 200 epochs but the plots zoom in on epochs 70 and onward for better visibility.

in CoE RAISE WP4. In light of this, the hypertuning of MLPF was used as an example workflow to benchmark HPO algorithms in Ray Tune by running a variety of them using four compute nodes. The number of samples drawn were varied and results were analyzed in terms of samples drawn, compute hours spent, best achieved validation loss and best improvement per compute resources spent. The results are presented in figures 3 and 4. Figure 3 shows that both Hyperband and ASHA significantly outperforms random search in terms of core-hours spent per sample. Comparing the runs using ASHA, the combination with Bayesian optimization adds some overhead compared to the combination with random search making the ASHA + random search combination perform best from this perspective. Figure 4 gives another point of view, where the validation loss is plotted against the core-hours spent. From this viewpoint, it is clear that ASHA + Bayesian optimization gives the highest improvement per spent core-hour.

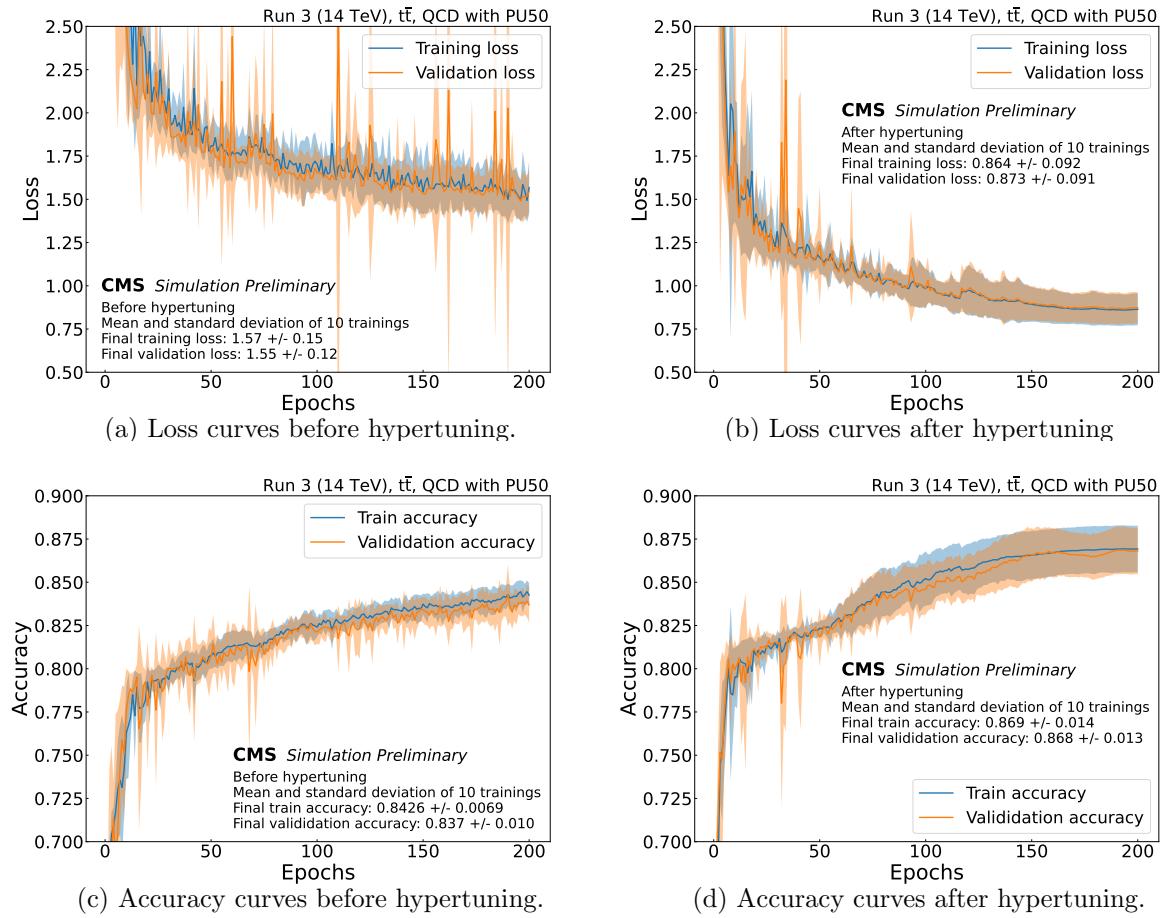


Figure 2: Mean and standard deviation of loss and classification accuracy as a function of the training epoch computed from 10 trainings.

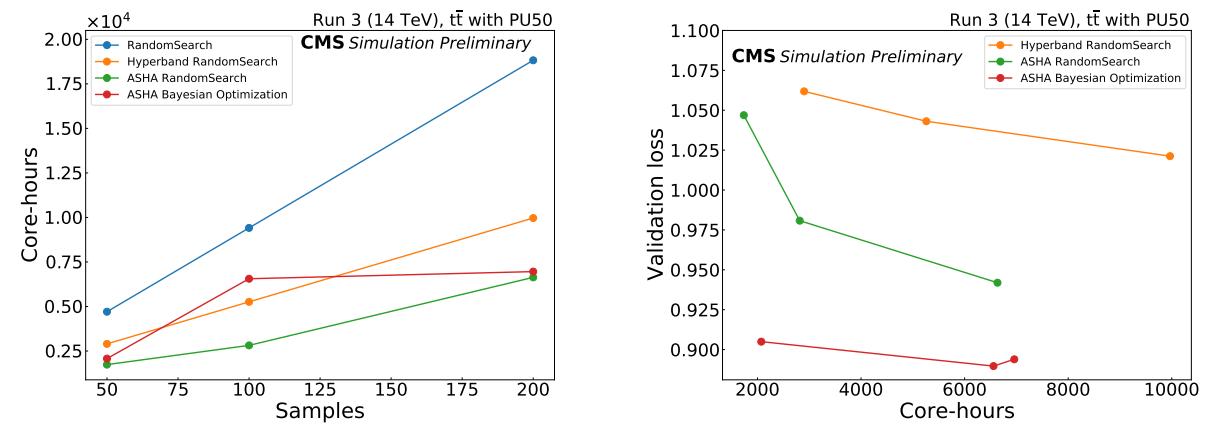


Figure 3: Comparison of hyperparameter search algorithms. Number of core-hours spent versus the number of samples drawn.

Figure 4: Comparison of hyperparameter search algorithms. Validation loss of the best trial versus number of core-hours spent.

#### 4. Conclusion

CoE RAISE develops novel, scalable AI methods towards Exascale with use cases from a wide range of sciences and industry. HPO could benefit any data-driven AI-based algorithm and in the example use case of MLPF, large-scale distributed hypertuning significantly increased model performance. This would not have been possible without access to HPC resources since it would have taken approximately half a year of continuous hypertuning on a single GPU. Other sciences and use cases in CoE RAISE are also adopting HPC for hypertuning, including the use cases of WP4, within fields such as remote sensing, seismic imaging, defect-free additive manufacturing and sound engineering.

#### Acknowledgments

We thank our colleagues in CoE RAISE, in particular Andreas Lintermann, Morris Riedel, Marcel Aach, Eric Michael Sumner, Eray Inanc, Michael Bresser, Jennifer Lopez Barrilao, Ieva Timrote and Christina Bolanou for helpful discussions and feedback in the course of this work. We also thank our colleagues in the CMS Collaboration, especially Javier Duarte, Farouk Mokhtar, Jieun Yoo, Jean-Roch Vlimant and Maurizio Pierini for their contributions to MLPF.

Eric Wulff was supported by CoE RAISE and Joosep Pata was supported by the Mobilitas Pluss Grant No. MOBTP187 of the Estonian Research Council. The CoE RAISE project has received funding from the European Union's Horizon 2020 – Research and Innovation Framework Programme H2020-INFRAEDI-2019-1 under grant agreement no. 951733.

#### References

- [1] The CoE RAISE project 2022 The CoE RAISE Website URL <https://www.coe-raise.eu>
- [2] Pata J, Duarte J, Vlimant J R, Pierini M and Spiropulu M 2021 MLPF: Efficient machine-learned particle-flow reconstruction using graph neural networks *Eur. Phys. J. C* **81** 381 (*Preprint* 2101.08578)
- [3] The CMS Collaboration *et al.* 2008 The CMS experiment at the CERN LHC *J. Instrum.* **3** S08004–S08004
- [4] Moritz P, Nishihara R, Wang S, Tumanov A, Liaw R, Liang E, Paul W, Jordan M I and Stoica I 2017 Ray: A distributed framework for emerging AI applications *CoRR* **abs/1712.05889** (*Preprint* 1712.05889)
- [5] Liaw R, Liang E, Nishihara R, Moritz P, Gonzalez J E and Stoica I 2018 Tune: A research platform for distributed model selection and training *CoRR* **abs/1807.05118** (*Preprint* 1807.05118)
- [6] Abadi M *et al.* 2015 TensorFlow: Large-scale machine learning on heterogeneous systems software available from tensorflow.org URL [https://www.tensorflow.org/](https://www.tensorflow.org)
- [7] Paszke A *et al.* 2019 Pytorch: An imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems* 32 ed Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E and Garnett R (Curran Associates, Inc.) pp 8024–8035
- [8] Head T *et al.* 2019 scikit-optimize <https://github.com/scikit-optimize/scikit-optimize>
- [9] Bergstra J, Yamins D and Cox D 2013 Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures *Proceedings of the 30th International Conference on Machine Learning (PMLR)* vol 28 ed Dasgupta S and McAllester D (Atlanta, Georgia, USA: PMLR) pp 115–123
- [10] Akiba T, Sano S, Yanase T, Ohta T and Koyama M 2019 Optuna: A next-generation hyperparameter optimization framework *CoRR* **abs/1907.10902** (*Preprint* 1907.10902)
- [11] Clark S and Hayes P 2019 SigOpt Web page <https://sigopt.com> URL <https://sigopt.com>
- [12] Sirunyan A M *et al.* 2017 Particle-flow reconstruction and global event description with the CMS detector *J. Instrum.* **12** P10003–P10003 (*Preprint* 1706.04965)
- [13] Pata J for the CMS Collaboration 2022 Machine learning for particle flow reconstruction at CMS *J. Phys.: Conf. Series* **These proceedings**
- [14] Pata J, Wulff E, Mokhtar F, Duarte J and Tepper A 2021 jpata/particleflow: Baseline MLPF model for CMS DOI 10.5281/zenodo.5520559 URL <https://github.com/jpata/particleflow>
- [15] Jülich Supercomputing Centre 2021 JUWELS Cluster and Booster: Exascale Pathfinder with Modular Supercomputing Architecture at Juelich Supercomputing Centre *JLSRF* **7**
- [16] Falkner S, Klein A and Hutter F 2018 BOHB: robust and efficient hyperparameter optimization at scale *CoRR* **abs/1807.01774** (*Preprint* 1807.01774)
- [17] Li L, Jamieson K G, Rostamizadeh A, Gonina E, Hardt M, Recht B and Talwalkar A 2018 Massively parallel hyperparameter tuning *CoRR* **abs/1810.05934** (*Preprint* 1810.05934)

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/361742207>

# Extended Abstract: Making AI work for skills-based training: A case study

Conference Paper · April 2022

---

CITATIONS

0

READS

39

8 authors, including:



Robby Robson

Eduworks

84 PUBLICATIONS 635 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Intelligent Tutoring Systems [View project](#)



Knowledge Management [View project](#)

## Extended Abstract: Making AI work for skills-based training: A case study

R. Robson<sup>1</sup>, E. Kelsey<sup>1</sup>, A. Goel<sup>2</sup>, L. Egerton<sup>1</sup>, S. Nasir<sup>1</sup>, M. Lisle<sup>2</sup>, A. LaFleur<sup>1</sup>, and E. Robson<sup>1</sup>

<sup>1</sup>Eduworks Corporation, USA. roby.robson@eduworks.com. <sup>2</sup>Georgia Institute of Technology, USA.

**Abstract** — Motivated by the need to upskill workers rapidly and equitably in response to changes in work and the workplace, a team funded by the US National Science Foundation Convergence Accelerator program has developed and piloted an application called *SkillSync* connects companies to college non-degree programs and facilitates the exchange of training requirements and relevant training opportunities. This abstract describes *SkillSync* and how Artificial Intelligence (AI) services are used in *SkillSync* to automate skills extraction and align training resources with required skills. This abstract then identifies issues we encountered in developing *SkillSync* and discusses lessons learned. The last section discusses further applications.

### 1 Introduction

The work presented here is supported by the US National Science Foundation (NSF) [1] and is motivated by the need to upskill millions of workers in response to changing job requirements. Since non-degree programs at local colleges are a natural but under-utilized source for training, we are developing an AI-enabled web app called *SkillSync* that connects companies to colleges for upskilling workers. *SkillSync* has undergone extensive design exercises with focus groups and been successfully used in three live trials, with additional pilots scheduled through 2022.

*SkillSync* takes a skills-based approach that requires the ability to identify and compare knowledge, skills, and abilities (KSAs) [2] in unstructured text. We have applied recent advances in natural language understanding (NLU) and machine learning (ML) to develop *AI services* that do this. These are exposed through application programmer interfaces (APIs) that are used by the *SkillSync* app and are available to other applications. This abstract describes the app, lists the AI services, identifies the issues we faced and overcame, and ends with takeaways and applications.

### 2 The SkillSync App

Our initial goal was to bridge the gap caused by the rapid pace at which industry needs evolve and the much slower pace at which academic programs operate and change. To understand the problem more deeply, we met with human resources (HR) and talent managers, directors of college continuing and professional education programs in the Atlanta area, the Business Higher Education Forum and the University Professional and Continuing Education Association. Our findings led to the development of an app that digitizes the connection between employers and colleges for the purpose of upskilling incumbent workers.

This app has two user types of users: *company users* responsible for HR, talent development and training at a company and *college users* responsible for coordinating (non-degree) training programs with company customers. Company users create *training requests* that identify skills to be acquired, together with information about the existing skills potential trainees possess. They then publish requests to specific colleges or to a marketplace. College users receive these requests, create *training proposals* that respond to them, and send them (via *SkillSync*) to company users for review and action [3].

Company users can search skills from many external sources (e.g., O\*NET) by job title or keyword. They can prioritize skills, add new skills, and automatically extract skills from job descriptions they upload. In the Minimum Viable Product (MVP) release, users will be able to see skills trends derived from millions of live job postings from the National Labor Exchange (NLx) [5]. College users can import course data and search and select training offerings. When a college user selects a set of course offerings, *SkillSync* displays an *alignment score*, a number between 0 and 100 that indicates how well the selected offerings cover the skills in a training request. *SkillSync* also shows users how adding new offerings will affect the alignment score. An intelligent agent called *AskJill* provides help and answer questions about the app. The *SkillSync* web site includes up-to-date feature lists and to view videos that show *SkillSync* in action [6].

### 3 AI Services

*SkillSync* uses five AI services: *KSA Extraction* identifies Knowledge, Skills, and Abilities (KSAs and relationships among them in unstructured text); *KSA Generation* generates a prioritized list of the most relevant KSAs for a job title or description; *Alignment* computes an alignment score between a KSA and a course description by matching explicit and latent concepts, *CII Removal* replaces *Company Identifiable Information* (company names, brands, trademarks, technologies, processes, people and locations) with generic equivalents; and *AskJill* interprets and answers use questions about *SkillSync* in real time via a text interface. The first three are core to the operation of *SkillSync* and to skills-based talent management. The fourth is required to meet commercial requirements for accessing job data but is applicable to other use cases, such as scrubbing military personnel records. The fifth is an active area of research with the potential to increase the usability and trustworthiness of AI-enabled training and talent management applications.

All AI services use large, pre-trained language models such as BERT [7] and GPT-2 [8] as starting points, with transfer learning used to fine-tune pre-trained language models to perform specific tasks in specific domains. In CII Extraction, transfer learning was applied to an off-the-shelf pre-trained BERT model. The first three are based on language models pre-trained from scratch with data drawn from Wikipedia, a curated subset of the Common Crawl database of web crawl data, a database of job postings

provided by the National Labor Exchange, and a dataset consisting of curated, open-source textbooks, course descriptions, and training materials.

AskJill is being developed at the Georgia Institute of Technology (Georgia Tech) by the Design & Intelligence Laboratory at [9]. It is based on technology that analyzes and answers student questions about courses based on the content of syllabi [10] and that has been used to explain the design and operation of an interactive tutoring system [11]. AskJill uses a two-dimensional hybrid ML and semantic processing model. An NLP-based model is trained to classify the *intent* of a question. A semantic processing layer converts intent into a structured query, searches data in structured and unstructured knowledge bases, and formulates a natural-sounding response.

#### 4 Issues Encountered

Several issues were encountered (and overcome) in developing AI services for SkillSync. The first was bias. Language models can reflect gender, ethnic, racial and other biases inherent in the sources used to train them [12]. This has been observed in job descriptions [13]. In response, we used multiple methods to reduce bias, as measured by how closely job related terms are to racial or gendered terms in embeddings, whether racial or gendered terms co-occur with an occupation, and whether there are associations between race or gender, an occupation, and positive or negative sentiment. Our bias reduction methods have reduced these measures, which is encouraging.

A second issue is data acquisition. Our language models require labelled training data for each occupational domain. Subject matter expertise is needed to label the data, and data labelling is time consuming and repetitive. This makes it hard to find appropriate labor. After trying alternatives, we turned to commercial data labelling firms. These were more costly and had longer turnaround times than anticipated, and the results often required re-working. A related issue was that most existing skills frameworks are in non-machine-actionable formats (PDF, Word™, or HTML) and the skills in these frameworks are often too high level or too contextualized to be useful. We spent considerable effort finding and curating skills frameworks and converting them into machine-actionable data.

The time required for AI service development was also an issue. ML pipelines can configure and automate repeatable execution of model and generation [14], but the tools that these are complex and still evolving. In practice, they need specialized knowledge to operate and can be fragile. Significant collaboration among data scientists, data engineers, and IT is needed to configure, optimize and debug ML pipelines and the cloud-based computational environments used for model training. The computations themselves can take days and often cannot be parallelized. This slowed the development of AI services.

The alignment score was another issue. We wanted this score to reflect the match between prioritized set of KSAs and a set of course materials, but we knew of no theory that rigorously defined this match. As a result, we took an empirical approach in which success was gauged by user acceptance and whether the score behaved as

expected when course materials were added or removed. The score we used in trials passed the “sniff test” in that it increased and decreased in a logical fashion and the results made sense to users, but more work is needed to ground the alignment in theory.

Another issue arose from a contractual requirement. SkillSync analyzes KSAs in job postings from the NLx to detect skills trends. Use of these postings is governed by an agreement that requires removing CII from job postings and KSAs to avoid revealing competitive information. This is a challenging NLP problem that we solved with algorithms that are layered on a version of named entity recognition (NER) [15] that incorporates *attention* [16].

The last issue concerns AskJill, which was first designed to answer questions about course materials. In focus groups, we discovered that company were interested in seeing the actual course descriptions, for which we did not need AskJill. AskJill is now designed to answer questions about the app, its operation, and its algorithms with the goal of improving transparency, explainability, and usability [17] (as well as providing contextual help). This shift requires more sophisticated knowledge-based reasoning, which is being developed using Task-Method-Knowledge (TMK) models that Georgia Tech has used for meta-reasoning in a variety of AI Systems [18] [19].

#### 5 Applications and Lessons Learned

SkillSync and its AI services can support a broad range of applications. The techniques used for CII removal can be used to redact sensitive documents, we are exploring how to use the app to help military and government personnel find voluntary education, and KSAs collected from NLx will be used to identify skills trends and in-demand skills. This information will be provided to company and college users and disseminated in the NLx research hub [20]. AskJill is intended to be a generalizable asset that can be trained to provide help and increase trust in other apps.

In addition to producing generalizable technology, our work has taught us two important lessons that apply to anyone who wishes to develop AI-enabled skills-based training and talent management applications. The first is that while AI may be supported by ML packages, pipeline tools, and pre-trained models, there are factors (such as data acquisition) that can lead to unanticipated costs, and it is not unusual to encounter non-standard challenging requirements (such as bias reduction and CII removal in our case). The need for in-depth knowledge of ML, NLU, and related data science and engineering processes should not be underestimated. The second lesson is that apps like SkillSync operate in sociotechnical environments where technology and human behavior are intertwined. We made numerous pivots in the design of the app and its underlying AI services based on focus groups and trials and real-world business requirements, and we continue to fine tune many aspects as we engage with more and more diverse users. Underestimating the need for end-user input is a more fatal error than underestimating the complexity of the problems faced in transitioning to skills-based talent management and in using AI to support this transition.

## 6 References

- [1] NSF, *Award #2033578*, 2020.
- [2] US Veterans Administration, “What Are KSAs? - VA JOBS,” [Online]. Available: <http://www.va.gov/jobs/hiring/apply/ksa.asp>. [Accessed 27 December 2021].
- [3] BHEF, “The Business Higher Education Forum,” [Online]. Available: <https://www.bhef.com/>. [Accessed 27 December 2021].
- [4] UPCEA, “About UPCEA,” [Online]. Available: <https://upcea.edu/about/>. [Accessed 27 December 2021].
- [5] NLx, “National Labor Exchange,” [Online]. Available: <https://usnlx.com/>. [Accessed 27 December 2021].
- [6] Eduworks Corporation, “SkillSync,” [Online]. Available: <https://www.skillsync.com/>. [Accessed 27 December 2021].
- [7] J. Devlin, M.-W. Shang, K. Lee and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, Google, 2019.
- [8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, “Language models are unsupervised multitask learners,” 2019. [Online]. Available: [https://cdn.openai.com/better-language-models/language\\_models\\_are\\_unsupervised\\_multitask\\_learners.pdf](https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf). [Accessed 28 December 2021].
- [9] Georgia Tech, “Design & Intelligence Lab,” [Online]. Available: <https://dilab.gatech.edu/>. [Accessed 08 January 2022].
- [10] A. Goel and L. Polepeddi, “Jill Watson, A virtual teaching assistant for online education.,” in *Education at Scale: Engineering Online Learning and Teaching.*, C. Dede, J. Richards and B. Saxberg, Eds., Routledge, 2018.
- [11] A. Goel, V. Nandan, E. Gregori, S. An and S. Rugaber, “Explanation as Question Answering based on User Guides,” in *AAAI-2022 Workshop on Explanation in Agency*, 2022.
- [12] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama and A. T. Kalai, “Man is to computer programmer as woman is to homemaker? debiasing word embeddings,” *Advances in neural information processing systems*, vol. 29, pp. 4349-4357, 2016.
- [13] H. R. Kirk, F. Volpin, E. Benussi, F. Dreyer, A. Shtedritski and Y. Asano, “Bias Out-of-the-Box: An Empirical Analysis of Intersectional Occupational Biases in Popular Generative Language Models,” *Advances in Neural Information Processing Systems (NeurIPS 2021)*, vol. 34, 2021.
- [14] A. Barrak, E. E. Eghan and B. Adams, “On the Co-evolution of ML Pipelines and Source Code-Empirical Study of DVC Projects,” in *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Virtual, IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER).
- [15] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Linguisticae Investigationes*, vol. 30, no. 1, pp. 3-26, 2007.
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, N. A. Gomez, L. Kaiser and I. Polosukhin, “Attention is all you need,” in *31st Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, 2017.
- [17] K. Wiggers, “What is explainable AI? Building trust in AI models,” VentureBeat, 26 November 2021. [Online]. Available: <https://venturebeat.com/2021/11/26/what-is-explainable-ai-building-trust-in-ai-models/>. [Accessed 27 December 2021].
- [18] W. Murdock and A. Goel, “Meta-Case-Based reasoning: Self-Understanding for Self-Improvement.,” *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 20, no. 1, pp. 1-36, 2008.
- [19] A. K. Goel and S. Rugaber, “GAIA: A CAD-Like Environment for Designing Game-Playing Agents,” *IEEE Intelligent Systems*, vol. 32, no. 3, pp. 60-67, 2017.
- [20] NLx, “Welcome to the NLx Research Hub,” National Labor Exchange, [Online]. Available: <https://nlxresearchhub.org/>. [Accessed 29 December 2021].

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/366500774>

# THE AIS HAVE IT? HACKING INTO THE AI AVATAR DREAM

Preprint · December 2022

DOI: 10.13140/RG.2.2.22539.77605

---

CITATIONS

0

READS

94

1 author:



Alexander Gerner

University of Lisbon

70 PUBLICATIONS 18 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Philosophy of Cognitive Enhancement [View project](#)



Image in Science and Art [View project](#)

# THE AIS HAVE IT? HACKING INTO THE AI AVATAR DREAM<sup>1</sup>

ALEXANDER GERNER

## 1. TOWARDS THE AI AVATAR DREAM

Avatars, artificial persons, or graphic placeholders for human beings are used in various functions in today's *cultures of digitality*. Avatars range from cartoon figures – starting with "Clippy," the famously annoying Microsoft Word paperclip assistant whose googly eyes watch our moves on the screen – to virtual workforce employees, social partners, and programmed AI therapists. With avatars, we have to heed the disappearance of computers in society in the quest for digital humanity (Simanowski 2019: 3) by criticizing mere data-driven media and their cultural analytics (Manovich 2020) as models of AI avatar aesthetics.

The avatar as a model of subjectivity has been described as a virtual proxy and representative of a real person (Little 1999; cf. Gunkel 2010). Others focus on a prosthetic avatar as a puppet or homunculus double (Apter 2008) of agency in a technical milieu, including cybertherapy (Gerner 2020).

The *avatar dream* (Fox Harrell and Lim 2017), when integrated with the two other culturally shared visions of future media of technological dreams using the computer and algorithms – the *smart dream* of ubiquitous quantitative total availability (Emrich / Roes 2011: 8–9) and the *AI dream* – becomes, in my view, the *smart, ubiquitous AI avatar dream*.

Fox Harrell and Lim characterized the avatar dream in a twofold way: technical and experiential. Computationally created surrogates engage us using text descriptions in games or social media through virtual visual representations in virtual reality environments. The experiential dimension enables virtual surrogate selves to engage in immersive experiences beyond orthodox physical encounters (Fox Harrell / Lim 2017: 52).

In this conception of the avatar dream, people utilize the computer as a chimera-creating tool to hack into their self-image. The avatar dream machine produces surrogates and

### IMAGE 1.

CLIPPY: IN OFFICE VERSIONS 97 AND 2000, IF A USER TYPED "DEAR" AT THE BEGINNING OF A DOCUMENT, CLIPPY WOULD APPEAR IN THE BOTTOM RIGHT CORNER OF THE SCREEN WITH A TEXT BUBBLE THAT READ, "IT LOOKS LIKE YOU'RE WRITING A LETTER. WOULD YOU LIKE HELP?"

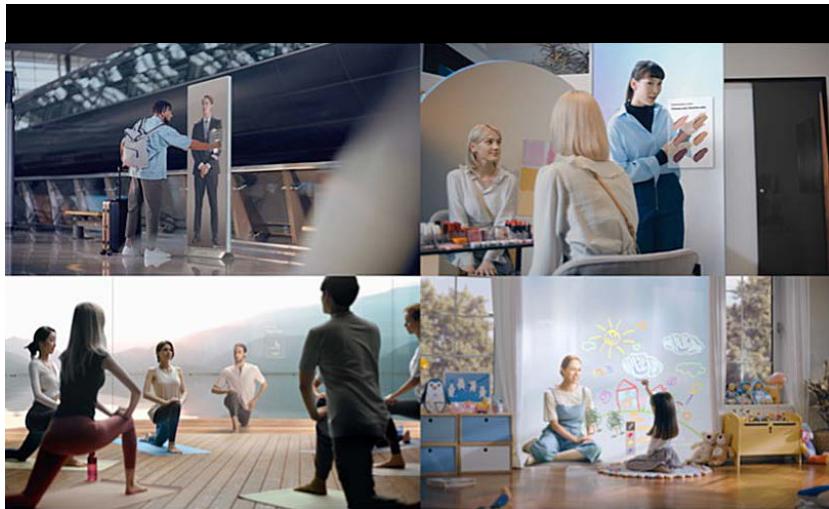
SEE FELDMAN (2016) ON THE DESIGN OF A VIRTUAL ASSISTANT OPTIMIZED FOR FIRST USE OF A FUNCTION THAT WAS THE FORERUNNER OF AI ASSISTANTS SUCH AS ALEXA OR SIRI  
[HTTPS://MONEY.CNN.COM/GALLERIES/2009/TECHNOLOGY/0910/GALLERY.MICROSOFT\\_WINDOWS\\_GAFFES/2.HTML](https://MONEY.CNN.COM/GALLERIES/2009/TECHNOLOGY/0910/GALLERY.MICROSOFT_WINDOWS_GAFFES/2.HTML)



<sup>1</sup> This research is financed by Portuguese national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., within the scope of the Transitional Standard – DL57/2016/CP CT[12343/2018], in the scientific field of History and Philosophy of Science and Technology, Project: *Hacking Humans. Dramaturgies and Technologies of Becoming Other*. Position: 2404.

transforms each imaginary experiential frame virtually by allowing us to play whoever we want to be. Thus, avatars might become part of a virtual identity. Fox Harrell and Lim (2017: 60) further argue that the avatar dream needs be reimagined beyond mere techno-phenomenological otherness to take into account society, including biases and stereotypes and constraints to the achievement of social identity as experienced in physical and self-imaginings in virtual worlds. Suppose we do not heed the historical, social, and cultural constraints of human-made artifacts. In that case, we might not avoid system-embedded and user-embedded “box effects” – “the experiences of people that emerge from the failure of classification system (...) stereotypes, social biases, stigmas, discrimination, prejudice, racism, and sexism” (Fox Harrell / Lim 2017: 54) – that would render the avatar dream impossible. While the avatar dream is specifically related to personal self-image, the AI avatar dream goes beyond a mere computational representation of users. Beyond mere mechanical “learning” or “intelligence,” the *AI avatar dream* proposes AI avatars as creative machines (Rauterberg 2021). AI avatar dreams create other AI personas and professional specialists (e.g., therapists or consultants), such as embodied cognitive models, and a dream of another vision of humanity. in which the avatar is even part of a future self-generating art. This AI avatar

dream goes in the direction of another artificial, virtual, or synthetic human: a form of self-superation, self-determination, and religious eternal self-salvation, with posthuman capacities, embodiment possibilities, and new modes of an extended human experience. Thus, virtual AI humans generate the



FUTURE OF WORK AND EVERYDAY LIFE” IN FOUR SITUATIONS, CLOCKWISE FROM TOP LEFT: AI AVATAR SERVICE ASSISTANT AT THE AIRPORT, AI AVATAR SALES AND SERVICE ASSISTANT, AI AVATAR TEACHER, AI AVATAR FITNESS TRAINER.

[HTTPS://NEON.LIFE/NEWS/CES-2021-PRESS-RELEASE](https://neon.life/news/ces-2021-press-release)

future media ability to communicate in natural human language, to “learn,” “remember,” and “own” a personality as well as making decisions by taking actions with their bodies via a set of sensory systems. The idea of virtual humans includes the ability to detect sensations, appraise sensation triggers, and respond to them.

AI avatars act as a digital workforce and function as employees, such as the virtual worker *AMELIA*. *AMELIA* is a job-based, human-equivalent digital employee that is customizable for each service business, such as for Customer Care, IT, and HR services or multi-lingual digital banking. In the case of the Sterling National Bank, *AMELIA* – renamed “Skye” – provides human-like communication and collaboration with the bank’s contact center agents and in the case of the Netherlands-based IT Service “Centric Burgerzaken” *AMELIA* is used to provide conversational AI, available 24/7, for digital public services for local government organizations.

AI avatars as workers are meant to enhance employee performance culture in VR scenarios within performance analytics. This development includes companies such as

Talespin's co-pilot virtual human training technology or customer assistance and UneeQ's Digital Humans, defined as AI-powered, lifelike virtual beings. UneeQ's Digital Humans are AI avatar workers that mimic human facial expressions, tone of voice, and body language in multimodal embodied forms of communication. These features are more important than mere language-based verbal communication for customer service. The abilities of virtual humans include showing emotion and different moods, making plans, and achieving goals, ideally set by some "internal" motivation. Internal motivation in the sense of Artificial General Intelligence (AGI) could even be an internal avatar model with an external avatar body – with the AGI ability, in addition to reasoning and problem-solving, to mimic the capacity of imagination and creativity. Burden and Savin-Baden (2019: 13) have developed a matrix to analyze virtual humans' traits on different spectra between self-aware and not self-aware, embodied and disembodied, humanoid and non-humanoid, natural-language and command-driven, autonomous and controlled, emotional and unemotional, personality-driven and impersonal, reasoning and unreasoning, learning and "unlearning" (cf. the EmoCOG architecture (Lin et al. 2011) or the OpenCOG architecture (Goertzel et al. 2014)

in which attention-related "forgetting" and memory resource management is put forward (Burden / Savin-Baden: 125)), and finally, imaginative and unimaginative. In a posthuman avatar case scenario, such as in Soul Machine's 4th and 5th level of AI avatars, the aims are not only spatial context, as-if imagination, and as-if intentionality, but also creative machine behaviors based on "learned experience" and "agency" for making discoveries and setting new intentions, plans, and goals. Moreover, AI avatars in the future AI dream world gain the ability to train themselves through interaction with humans and non-human systems. Finally, self-awareness and contextual understanding would emerge in independent digital, artificial persons with a strong semantic or contextual understanding of the AI avatar self's actions to create non-linear storytelling. Nevertheless, AI artifacts that move, speak, reason, and show radical mimetism will inevitably face issues of animism.

## 2. THE AIS HAVE IT? ON AI AVATARS

### 2.1 "HIGH FIDELITY" AVATARS: COUNTERFEIT OF HUMAN GAZE OR THE WRONG KIND OF ANTHROPOMORPHISM?

AI artifacts are AI systems that humans create for the purpose of radical mimesis: AI systems mimic actors who grant social faciality to machines in a way that seems human to observers. The AI avatar machine evokes movements of gaze and interest, as well as curiosity, and has to be critically assessed when reflecting on the topic of human or machine creativity. Coeckelbergh (2021) argues for a critical posthumanist point of view towards the anthropomorphism in technical objects that interact with humans. Should we then reject normative anti-anthropomorphism as nonsensical in social robotics and AI avatars? And still: we have to ask how we handle AI avatars not only as extensions of the self but as AI technology for human exploitation and data extraction (Crawford 2021), the cost of which must still be counted in its material, energetic and ecological aspects. Some may make a strong stance against AI avatars as simulation machines of not only intelligence but – foremost – human attributes such as creativity, autonomy, affectivity, and for being "artifactors," AI artifact systems that mimic human (like) actors, calling them a "counterfeit" (Pasquale 2020) of humanity. Therefore, the task of clearly separating AI systems from AI actors that mimic humans through anthropomorphic design stances might seem a good idea for a policy option (Cf. EU 2021) that calls for a renewal of Asimov's Three Laws of Robotics (see

Pasquale 2020: 3–19). These new potential rules for AI would go beyond avoiding malefice by impeding human substitution, human manipulation/counterfeiting, an AI arms race, and non-identification of artificial systems.

## 2.2 FROM AI CHILD AVATAR TO PLAYING GENERAL ARTIFICIAL INTELLIGENCE WITH A TOY CHILD MODEL: ON SOUL MACHINES'S AUTONOMOUS ARTIFACTOR ANIMATION

The AI research of the company *Soul Machines* “started with a baby”, called “Baby X” (Soul Machines 2021). According to IBM (*Soul Machines*. IBM. n.d.) and its Watson assistant integrated into Soul Machines, the aim and business challenges are to build on the paradoxical goal of *empathic AI* that has been staged as evolutionary human progress at the World Economic Forum in 2019 (Mantas 2019). The AI avatar model of Baby X plays interactively with the world around it, pragmatically making discoveries by manipulating things in the way we do. Animation stands at the center of Soul Machines’s business, which is inspired by the following questions: “What is the essence of animation? What if a character could autonomously animate itself and you could interact with it? How do you bring a digital character to life?” Baby X interacts with its surroundings by playing as if it were a child that learns, evolves, or “grows” its information base by testing the results of the games it plays; but does it actually rehearse and acquire reality? Soul Machines poses challenges of “problems to solve” that lie at the core of AI avatars as artifacts:

*How would we create biologically inspired artificial intelligence? And, build a digital consciousness to create affective computing that interprets and simulates human emotion, engaging autonomously? (Soul Machines 2020)*

Soul Machines’s AI avatar initially reminds us of an AI Tamagotchi (virtual pet), referring to the emotional annoyance of having to feed and care for the digital toy in how it is presented. However, the company aims to “make machines and AI as lifelike as possible,” envisioning “humanlike AI that has flexible intelligence and a dynamic interface that can relate to people”: human-AI relations seem to change in the age of machine learning, having a clear roadmap of how to achieve the highest levels of “autonomous animation.”

Soul Machines’s white paper (2020) distinguishes six stages of autonomous animation, in which level 0 and level 1 are dedicated to actually existent simulated, actor-driven, pre-recorded video or motion capture in which motion-capture cameras function as enabling technology for “possible solutions” in movie and games characters. On this level, avatars are supposed to be used as masks and puppets and heed movement notations of kinetic digital renderings in capturing performance art inside a motion

capture imaginary (Karreman 2017) in creative industries, games, films, and contemporary dance. Avatar masks refer to a performer as a puppet master: the avatar mask can be seen as an initial new identity or as a mere puppet in an uncanny zone in between *something* and *nothing*. For Soul Machines (2020), on Level 1, basic pre-authored animation that is

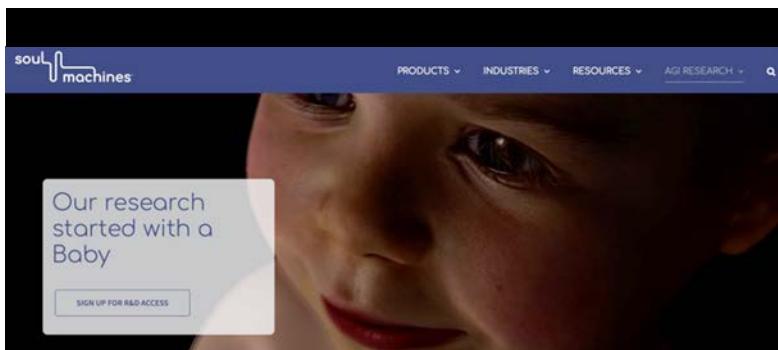


IMAGE 3.

### SCREENSHOT OF BABY X ON THE INTERNET PAGE SOUL MACHINES.

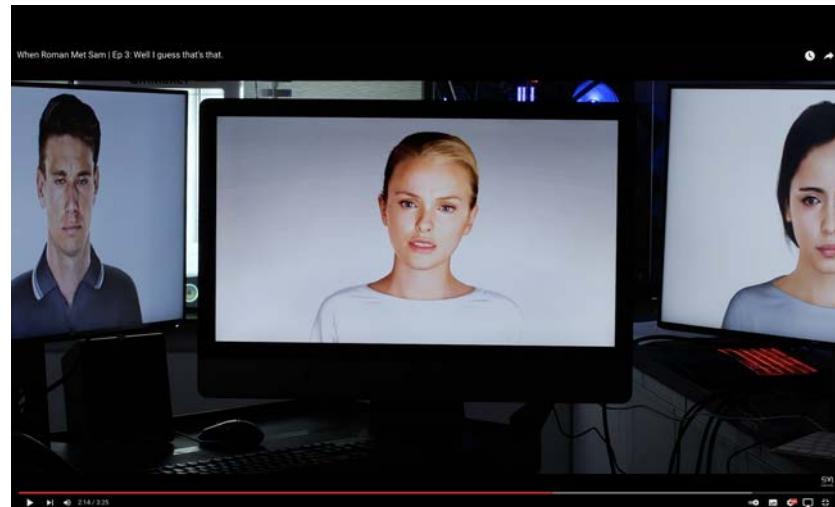
[HTTPS://WWW.SOULMACHINES.COM/RESOURCES/RESEARCH/BABY-X/](https://www.soulmachines.com/resources/research/baby-x/)

still actor-driven delivers pre-recorded movement based on simple triggers. The corresponding enabling technology would include the FAQ text-driven conversational database and pre-recorded voice content responses to create *digital puppets*. Levels 2 and 3 of “autonomous animation” would already use Natural Language Processing and “Dynamic Synthesized Human Behaviors” (Soul Machines 2020: 8), a “learning” capacity based as a solution on the Deep Fake level (level 2) or on level 3 with “[f]ull humanlike emotional responsiveness in facial animation including a conversational driven personality,” including on the voice level.

With the selling of the idea of the higher-level autonomous AI avatar as part of the AI avatar dream machine industry, we should ask: Does an AI avatar assimilate otherness by radical mimetics to be used in game design and performative conventions for creating *pervasive performances* (Peréz 2016: 16) between acting and engagement? Do AI avatars follow the metaphorical model of Turing’s child machine to create and provide “education” (Turing 2004: 460) to an AI child model such as Baby X or are they an AI avatar toy for playing around with artificial general

IMAGE 4.

intelligence, such as the AI toy avatar model Kanzi (Negarestani, 2018)?



**SCREENSHOT OF A TRIALOGUE OF THREE MACHINE COMMUNICATION AVATARS FROM SOUL MACHINES**

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=4MCDPFKYLTS](https://www.youtube.com/watch?v=4MCDPFKYLTS)

## 2.3 CODEC AVATARS (CA): THE QUEST TO PASS FACEBOOK’S “EGO AND THE MOTHER TEST”

As proximity and face-to-face encounters determine social relationships, the technological roadmap of VR and AR by Facebook’s Oculus Rift is heading towards overcoming distance and material barriers, as put forward by Tanaka, Nakanishi and Ishiguro (2014), who had shown that physical robot conferencing was superior to mere avatar chat. By virtual immersion of Codec Avatars, or enhanced Modular Codec Avatars (Chu et al. 2020) – which improve the robustness and expressiveness of traditional Codec Avatars – with holograms and VR/AR, Facebook aims at recreating and mimicking a sense of (artificial) VR telepresence, which provides remote and immersive telecommunication through VR headsets. The training phase of the VR telepresence system in the first stage is done by capturing facial expressions of a user with a multi-view camera dome and a VR headset for face modeling. In the final phase a personalized face animation model is derived using these correspondences, while the real-time photo-realistic avatar is driven from the VR headset cameras. This social teleportation is able to share eye gaze and expressive faciality that would be almost indistinguishable from the real-life presence of a person or object, even enhancing the spectrum of senses using a new artificial digital-media sense that could be called the digital immersive sense of foreshadowing proximity to an object or person.

However, Mark Zuckerberg admits that a) not all material experience while “connecting people” will and can be virtualized and b) algorithmically modeling the materiality of touch and haptics is not easily done. Photorealistic avatar models for “high-fidelity social interaction” of the users’ faces render avatars with a “Deep Appearance Model for Face rendering” (Lombardi et al. 2018) “using non-linear, photorealistic full-face models of geometry and texture” (Richard et al. 2020: 1), overcoming the shortcomings of mere geometric

approaches due to the “non-linearities in texture-based tongue motions and lip articulation” (*ibid.*). The difficulties are related to dark untracked geometry inside the cavities of the mouth that must be emulated with a synthetic texture of the mouth. Facebook came up with the idea that the avatars should not only be acceptable but also that they should not create uncanny valley effects. When setting up an avatar, a second “Turing Test” of social presence for the Facebook Codec Avatar would be if the avatar is acceptable for yourself and “your mother,” (Tech@Facebook 2019). Thus, Facebook focuses on their codec avatars as an avatar dream of a high-fidelity replica of the gaze. Implicit in Facebook’s High Fidelity Avatar (Schwarz et al. 2020: 91) is the concept of high fidelity of Skarbez et al. (2017), who differentiate between a) *physical morphological fidelity of looks* inside the operational environment, b) *functional action fidelity* of faciality of eye gaze or operational performance of the gaze in realistic movements and agency, and c) active *perceptive fidelity*.

However, I question if this hyperbolic-realistic “high fidelity” actually encompasses passive perception. What gives the face-to-face encounters a feeling of being together in the same space and experiencing a common “we”? Is it the idea of *being looked at by the other*, who does not perform exactly as I expect?



IMAGE 5.

### SCREENSHOT FACEBOOK CODEC AVATAR DEMONSTRATION VR TRAILERS AND CLIPS, YOUTUBE (JULY 2, 2020) “FACEBOOK’S PROTOTYPE PHOTOREAL AVATARS NOW HAVE REALISTIC EYES”

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=ETAMZMYKSG0](https://www.youtube.com/watch?v=ETAMZMYKSG0)

## 2.4 ERGOTIC COMMON-SENSE GESTURE-BASED AI AVATARS: TWENTY BILLION NEURON’S GESTURE SURROGATE AVATAR ASSISTANT MILLIE AND ITS AI APP FITNESS ALLY

The German/Canadian AI company Twenty Billion Neurons (TwentyBN), based in Berlin and Toronto, teaches machines to perceive like humans and developed the avatar “Millie” using situated a model of visual AI common sense via end-to-end learning on video clips (Twenty Billion Neurons 2020): the “Supermodel.” This AI model is a Python-based, deep learning gesture-recognition model based on large-scale crowd-acting operations and has collected millions of short video clips that require no depth information, as the model is entirely trained on 2D video data. This gesture recognition model internalizes a visual “common sense” of the world by identifying a wide range of fundamental human-object interactions and human body motions.

The TwentyBN avatar is based on the AI SuperModel of computer analysis of collected crowd-acting, in which people in the recorded video snippets perform common-sense hand control gestures via different data sets. These include, for example, Jester V1, in which 147 crowd workers performed 27 pre-defined hand gestures in front of a laptop camera or webcam (148,092 short clips of videos with different backgrounds, 3-sec length) and the “20BN-something-something V2 Dataset” inside the probability-guided labels to detect common-sense actions by AI algorithms of machine vision.

The neural network that offers the data feed to the avatar gesture simulation consists of short videos of mostly ergotic gestures sorted into common-sense pragmatic action classes (caption templates). These action classes are of a general “something [picking, moving, putting...] something” (Goyal et al. 2017: 5848) structure: AI avatars are based on common-sense gesture training sets fed into AI algorithms. These AI vision algorithms use artificial neural nets and deep fake technology. The avatar Millie is introduced in the first place as an interactive AI avatar in-store shopping assistant, and its corresponding app “Fitness Ally,” a virtual avatar fitness trainer, is used to guide the user through a series of workouts and to present them with recorded and interactive training data for fitness improvement.



IMAGE 6.

### SCREENSHOT TWENTYBN MILLIE'S FUNCTIONAL APPLICATION AS “DIGITAL IN-STORE EXPERT.”

SEE: [HTTPS://MEDIUM.COM/TWENTYBN/YOUR-DIGITAL-IN-STORE-EXPERT-FOR-EVERYTHING-D0865B82E27A](https://medium.com/twentybn/your-digital-in-store-expert-for-everything-d0865b82e27a).

THE SLOGAN OF THE COMPANY IS “BREATHING LIFE INTO VIRTUAL BEINGS/ OUR HUMAN-CENTRIC AI TECHNOLOGY BRINGS SEEING AND SOCIALE DIGITAL ASSISTANTS TO LIFE.” THE DATABASE IN 2017 CONSISTED OF MORE THAN 100,000 VIDEOS ACROSS 174 CLASSES; BY 2021 THE DATABASE HAD GROWN MORE THAN TENFOLD.



IMAGE 7.

### SMARTPHONE APP FITNESS ALLY “REAL-TIME INSTRUCTION AND MOTIVATION.”

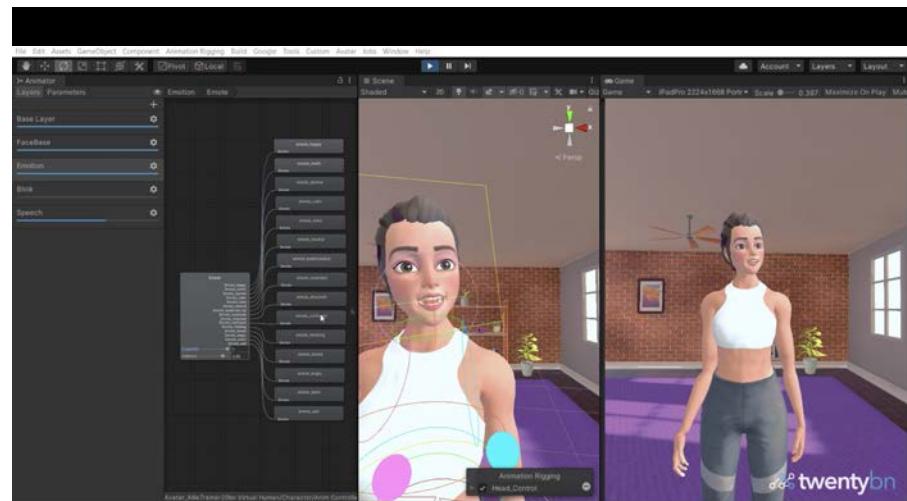
[HTTPS://FITNESSALLYAPP.COM](https://fitnessallyapp.com)

IMAGE 8.

### SCREENSHOT OF THE AI FITNESS TRAINER ANIMATION FACE STACK DESIGN MADE FOR ALLY FITNESS BY FERGUI.

([HTTPS://VIMEO.COM/543742486](https://vimeo.com/543742486)). THE TECHNICAL PROGRAMMING CHOSE KINETICS SKELETON BENCHMARKS FROM “PAPER-SWITHCODE” ([HTTPS://PAPERSWITHCODE.COM/SOTA/SKELETON-BASED-ACTION-RECOGNITION-ON-KINETICS](https://paperswithcode.com/sota/skeleton-based-action-recognition-on-kinetics)). THE POSE KEYPOINTS ARE DERIVED FROM THE “MMSKELETON” TOOLSET ON GITHUB ([HTTPS://GITHUB.COM/OPEN-MMLAB/MMSKELETON](https://github.com/open-mmlab/mmskeleton)).

IN GENERAL, THE FITNESS APP IS BASED ON THE ST-GCN MODEL THAT STANDS FOR “SPATIAL- TEMPORAL GRAPH CONVOLUTIONAL NETWORKS” (YAN ET AL. 2018), IN WHICH GRAPH CONVOLUTION IS TRANSPOSED TO SKELETON-BASED ACTION RECOGNITION AND ADDED THE MS-G3D MODEL TO CAPTURE COMPLEX SPATIAL-TEMPORAL FEATURES AS METHOD FOR IMPROVING SKELETON-BASED ACTION RECOGNITION BY MULTI-SCALE GRAPH CONVOLUTIONS AND A UNIFIED SPATIAL-TEMPORAL GRAPH CONVOLUTIONAL OPERATOR NAMED G3D (LIU ET AL. 2020), FOR ITS IMPLEMENTATION (TBN 2020, DEC 14: [HTTPS://MEDIUM.COM/TWENTYBN/PUTTING-THE-SKELETON-BACK-IN-THE-CLOSET-1E57A677C865](https://medium.com/twentybn/putting-the-skeleton-back-in-the-closet-1e57a677c865)). TWENTYBN IN 2020 ALSO LAUNCHED PART OF ITS TECHNOLOGY AS AN OPEN-SOURCE PLATFORM, SENSE, “A REAL-TIME ACTION RECOGNITION SYSTEM,” OPEN-SOURCE INFERENCE ENGINE FOR NEURAL NETWORK ARCHITECTURES THAT TAKES AN RGB VIDEO STREAM AS INPUT AND TRANSFORMS IT INTO A CORRESPONDING STREAM OF LABELS IN REAL TIME. SENSE INCLUDES DAY-TO-DAY HUMAN ACTIONS (PICKING UP OBJECTS, DRINKING WATER, FIXING YOUR HAIR, ETC.), HAND GESTURES, AND FITNESS EXERCISES, AMONG OTHERS: [HTTPS://GITHUB.COM/TWENTYBN/SENSE](https://github.com/twentybn/sense)



Millie was created with Deep Learning training of initially one thousand actions; now the database of common-sense gestures and visual common-sense actions to feed this action recognition pool is far over a million. It contains an object detector, an action/motion detector, a dialogue system, and a rule-engine for recognition and reaction to humans, which is used for Millie and was developed with the following aims (Kahn 2018): a) the immediate aim to build an interactive social sales assistant, gesture control systems for the car industry, and smart home devices b) the TBN long-term aim to build full digital avatars with a designed personality to interact with people in various settings, including full digital social companion, exploring avatars that could even “help” teach children in schools or instruct adults in skills such as yoga or cooking, or an artificial officer such as New Zealand’s police artificial person “Ella,” developed by Soul Machines. Whether AI avatars will attain the depth and personality to serve not merely as trainers but actually as pedagogic teachers is an issue remains to be seen.

## REFERENCES

- Apter, E. (2008), Technics of the Subject: The Avatar-Drive, in: Postmodern Culture, 18(2). doi:10.1353/pmc.0.0021
- Soul Machines (2021, March 8). Baby X. <https://www.soulmachines.com/resources/research/baby-x/> (04.02.2022).
- Brownlee, M. (2020, September 16) Youtube: Talking Tech and Holograms with Mark Zuckerberg! [https://www.youtube.com/watch?time\\_continue=72&v=eAagtcAup0&feature=emb\\_title](https://www.youtube.com/watch?time_continue=72&v=eAagtcAup0&feature=emb_title)
- Burden, D., / Savin-Baden, M. (2019), Virtual Humans: today and tomorrow. CRC PRESS.
- Chu, H. / Ma, S. / De la Torre, F. / Fidler, S. / Sheikh, Y. (2020), Expressive telepresence via modular codec avatars. Computer Vision – ECCV 2020, p. 330–345. doi:10.1007/978-3-030-58610-2\_20
- Crawford, K. (2021) Atlas of AI. New Haven.
- Emrich, H. / Roes, M. (2011), Engel und Avatar. Berlin.
- EU (2021), Proposal for a Regulation on a European approach for Artificial Intelligence: <https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-european-approach-artificial-intelligence> (26.4.2021)
- Feldman, B. (2016), Clippy Didn't Just Annoy You – He Changed the World, New York Magazine 31.10.2016, <https://nymag.com/vindicated/2016/10/clippy-didnt-just-annoy-you-he-changed-the-world.html> (12.12.2021)
- Fox Harrell, D./ Lim, C. (2017), Reimagining the Avatar Dream: Modeling Social Identity in Digital Media, in: Communications of the ACM, July 2017, 60 (7), p. 50-61.
- Gerner, A. (2020), Hacking into Cybertherapy: Considering a Gesture-enhanced Therapy with Avatars (g+TA). Kairos. Journal of Philosophy and Science 23, p. 32-87.
- Goyal, R. et al. (2017), The ‘Something Something’ Video Database for Learning and Evaluating VISUAL common sense. 2017 IEEE International Conference on Computer Vision (ICCV), p. 5842-5850. doi:10.1109/iccv.2017.622
- Gunkel, D. (2010), The Real Problem: Avatars, Metaphysics, and Online Social Interaction, New Media & Society 12(1), p. 127-141.
- Goertzel, B. / Hanson, D. / Yu, G. (2014), Toward a Robust Software Architecture for Generally Intelligent Humanoid Robotics. Proceeding Computer Science, 41, p. 158–163.
- Yan, S. / Xiong, Y. / Lin, D. (2018), Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. arXive: 180107455v2 [cs. CV] 25. January 2018
- Kahn, J. (2018), "Meet Millie the Avatar," Bloomberg News: <https://www.bloomberg.com/news/articles/2018-12-15/meet-millie-the-avatar-she-d-like-to-sell-you-a-pair-of-sunglasses> (14.07.2021)
- Lin, J. / Spraragen, M. / Blythe, J. / Zyda, M. (2011), EmoCog: Computational Integration of Emotion and Cognitive Architecture. In Proceedings of the Twenty-Fourth.FLAIRS Conference. Palo Alto, CA: AAAI.
- Liu, Z. / Zhang, H. / Chen, Z. / Wang, Z. / Ouyang, W. (2020), Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition. arXive:2003.14111v2 [cs.CV] 19 May 2020,; <https://arxiv.org/pdf/2003.14111.pdf> (20.07.2021)
- Little, G. (Fall 1999), 'A Manifesto for Avatars,' Intertexts 3(2): <http://www.gregorylittle.org/avatars/text.html> (20.04.2021)
- Mantas, J. (2019), Empathic AI Could be the Next Stage in Human Evolution - If We Get it Right. <https://www.weforum.org/agenda/2019/07/empathic-ai-could-be-the-next-stage-in-human-evolution-if-we-get-it-right/> (02.09.2020)
- Manovich, L. (2020), Cultural Analytics. Cambridge, Mass.
- Negarestani, R. (2018), Intelligence and Spirit. New York.
- Pasquale, F. (2020), New Laws of Robotics. Defending Human Expertise in the Age of AI. Cambridge Mass..
- Rauterberg, H. (2021), Die Kunst der Zukunft. Über den Traum von der kreativen Maschine, Berlin.
- Richard, A. / Lea, C. / Ma, S. / Gall, J. / De la Torre, F. / Sheikh, Y. (2020), Audio-and Gaze-driven Facial Animation of Codec Avatars: <https://research.fb.com/videos/audio-and-gaze-driven-facial-animation-of-codec-avatars/> 2020/ (03.02.2022)
- Schwartz, G. / Wei, S.-E. / Wang, T.-L. / Lombardi, S. / Simon, T. / Saragih, J. / Sheikh, Y. (2020), The Eyes Have It: An Integrated Eye and Face Model for Photorealistic Facial Animation. ACM Trans. Graph. 30, 4, Article 91.
- Simanowski, R. (2019), Stumme Medien. Vom Verschwinden der Computer in Bildung und Gesellschaft, Berlin
- Skabenz, R. / Brooks, F. / Whitton, M. (2017), A Survey of Presence and Related Concepts, ACM Computing Survey, 50, 6, Article 96, p. 1-39.
- SoulmachinesTM (2020), Delivering on the Promise of AI. How Digital People Rise Above Other Technologies. The Questions Every Decision Maker, Investor, and Innovator Should Ask, White paper, p. 1-12 [https://www.soulmachines.com/wp-content/uploads/2020/09/DeliveringOnThePromise.pdf?utm\\_medium=email&\\_hsMI=107119114&\\_hsenc=p2ANqtz--qePpUpMWU-NRc-1MaZa-u-i0-8NcC2fgomy8TrMthr-Y10iK\\_b1Z3nMkMizv8O9eag6JWsi33s-1R1R6z6G-7AfblA6ag&utm\\_content=107119114&utm\\_source=hs\\_automation](https://www.soulmachines.com/wp-content/uploads/2020/09/DeliveringOnThePromise.pdf?utm_medium=email&_hsMI=107119114&_hsenc=p2ANqtz--qePpUpMWU-NRc-1MaZa-u-i0-8NcC2fgomy8TrMthr-Y10iK_b1Z3nMkMizv8O9eag6JWsi33s-1R1R6z6G-7AfblA6ag&utm_content=107119114&utm_source=hs_automation)
- Tanaka, K. / Nakanishi, H. / Ishiguro, H. (2014), Comparing Video, Avatar, and Robot Mediated Communication: Pros and Cons of Embodiment. Communications in Computer and Information Science, p. 96–110. Doi:10.1007/978-3-662-44651-5\_9
- Turing, A. (2004), Computing Machinery and Intelligence (1950), in: Copeland, J. (Ed.), The Essential Turing. Oxford, p. 441-464.
- Twenty Billion Neurons (2020), Towards Situated Visual AI via End-to-End Learning on Video Clips, <https://medium.com/twentybn/towards-situated-visual-ai-via-end-to-end-learning-on-video-clips-2832bd9d519f> (10.02.2022)

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/369118966>

# Impact of AI on Retail Operation and Profitability Moderated by Employee Motivation and Upskilling

Article · March 2023

DOI: 10.2015/IJIRMF/202302028

---

CITATIONS

0

1 author:



Dr. L R K Krishnan PhD  
VIT Business School Chennai Tamil Nadu India

73 PUBLICATIONS 49 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Legal Studies [View project](#)



Attrition and Employee Turnover Studies [View project](#)



## Impact of AI on Retail Operation and Profitability Moderated by Employee Motivation and Upskilling

<sup>1</sup>Akshaya. V,      <sup>2</sup>Dr. L.R.K. Krishnan

<sup>1</sup>VIT University, Business school, Chennai. Email - akshayavinay.akvi@gmail.com

<sup>2</sup> VIT University, Business school, Chennai.

**Abstract:** This article provides a comprehensive overview of the digital transformation of the retail industry and describes the influences on employee motivation, upskilling, employee productivity and innovations that it offers in retail store performance. Artificial intelligence (AI) allows human work to be shifted toward technological systems that are currently not fully capable. Incorporating AI tools in employee training will have a profound impact on profitability in a business which will result in sustainability of the business. AI imparts companies a host of ways to better understand, predict, and engage customers. Furthermore AI tools has positive impact on employee motivation and upskilling. The objective if the research is to study the Impact of AI on employee training and upskilling which impact on the profitability in the retail business and enhances the sustainability in market. The methodology used was qualitative and included in-depth literature review and synthesizing observations and results made in field studies. Researchers have found direct links to various academic conversations surrounding this area of research. Using this framework retailer can have a better insight over the disruptive tools and their impact on employee motivation and upskilling. The findings showed that incorporation of disruptive tools in employee training has a significant impact on customer choices, preferences which enhances business sustainability.

**Key Words:** Disruptive Tools, Employee Motivation, Artificial Intelligence, Machine learning, Business, Sustainability.

### 1. INTRODUCTION:

Artificial intelligence (AI) is having a significant impact on the retail industry, transforming the way retailers operate and interact with customers. Artificial intelligence (AI) is increasingly being adopted in the retail sector, resulting in a significant impact on both retail operations and profitability. (Taguimdjé et al., 2022) AI covers a wide range of technologies, including machine translation, chatbots, and self-learning algorithms, all of which can allow individuals to better understand their environment and act accordingly. (Deo & Khedkar 2018), provides an overview of personalized product recommendation techniques in e-commerce, including the use of collaborative filtering, content-based filtering, and hybrid methods. However, these effects can be motivated through employee motivation and training. In retail, AI can streamline processes, automate tasks, and free up employees to focus on more value-adding activities. For example, AI-supported chatbots, handle customer inquiries and enable employees to provide more personalized customer service. Similarly, AI-powered supply chain management can improve inventory management, reduce the need for manual intervention, and reduce the risk of out-of-stock. Regarding profitability, AI can help retailers better understand customer preferences and behaviors, allowing them to personalize marketing efforts and increase sales. Additionally, AI-powered pricing algorithms help retailers dynamically set prices based on real-time supply and demand data, improving profitability. However, the impact of AI on retail operations and profitability can be improved by motivating and upskilling employees.

If employees are unmotivated to use new technology, they may resist implementing it, resulting in reduced efficiency and productivity. (Maity, S. (2019)) Training needs are becoming more personalized. Micro-learning and byte-sized training modules, easily accessible to employees, as and when required, are some of the major organizational needs. Training and development programs should be designed keeping in mind factors of employee engagement, involvement, and extent of training transfer. Additionally, if an employee is not trained to operate an AI system, the employee may struggle to use it effectively, resulting in poor performance and diminished profits. Therefore, to maximize the benefits of AI in retail, it is essential to motivate employees and provide them with the necessary training to enable them to use new technologies effectively. This includes regular training, clear communication, and a supportive work environment that encourages innovation and continuous learning.



## 1.1 AI in the retail operation:

Artificial intelligence (AI) is playing a significant role in transforming the way retail operations are managed and improving their overall efficiency. AI can help retailers optimize inventory levels, reduce stock outs and overstocking, and improve supply chain efficiency. AI-powered chatbots can handle customer inquiries and support, freeing up employees to provide more personalized customer service. Predictive Analytics can help retailers make informed decisions by analyzing data on customer behavior, market trends, and sales data. AI-powered visual search technologies can help customers find the products they are looking for by allowing them to search using images or videos. Fraud Detection can help retailers detect fraudulent activity in real time, reducing losses and improving security. AI-powered algorithms can analyze customer data to create highly personalized experiences and marketing campaigns, increasing customer engagement and loyalty. (Krishnan et al., 2022) technology-based (AI), where human constraints can be nullified. With this knowledge, they were able to expand their productivity. AI-powered pricing algorithms can dynamically set prices based on real-time demand and supply data, improving profitability. AI is enabling retailers to streamline their operations, reduce costs, and improve the customer experience. However, it's important to note that while AI has the potential to greatly benefit retail operations, it must be used ethically and responsibly to avoid potential negative consequences such as job loss or privacy violations.

## 1.2. Impact of AI on Employee Motivation and Upskilling:

The impact of artificial intelligence (AI) on employee motivation and upskilling is a complex and nuanced issue. On one hand, AI can automate repetitive and mundane tasks, freeing up employees to focus on higher-level, more fulfilling work. This can lead to increased motivation and job satisfaction. It's important for businesses to proactively invest in upskilling their employees to acquire new skills and remain relevant in the age of AI. This can include training programs in areas such as data analysis, software development, and digital marketing. Additionally, companies can take steps to foster a culture of continuous learning and professional development, encouraging employees to continuously develop their skills and stay up-to-date with new technologies. This can help employees feel more confident and secure in their jobs, leading to increased motivation and job satisfaction. Overall, while AI has the potential to greatly impact employee motivation and upskilling, businesses need to approach its integration into the workplace thoughtfully and proactively, taking steps to minimize potential negative impacts and support employee development.

## 2. LITERATURE REVIEW:

The retail sector is characterized in many countries by oligopoly markets with intense competition among incumbent retailers and increasing competition between traditional and new 'pure' digital players (Schutte, 2017). This increased competition has led to the need for caution to distinguish between facility types (Meffert et al., 2015), increased costs, and overall price awareness (Daurer et al., 2012). This has led to the impact of the company's price image on Selected retail chains. Therefore, companies must remain competitive. (Krishnan et al., 2022) Organizations are heavily investing in AI and ML tools and reaping the benefits, securing a competitive advantage. Emerging technologies are replacing human effort in information processing with considerably faster and more precise technologies, allowing corporate leaders to make faster and more consistent judgments. Complex analysis and decisions in price management can be performed with intelligent, self-learning solutions. Dynamic pricing (Kephart et al., 2000) is a new development in pricing strategies in which companies adjust the price of their products and services to current market demand in real-time. AI is used as an automatic algorithm to calculate prices. Human decisions cannot keep up with the speed required and the amount of data to consider (Jaekel, 2017). AI is also used to customize store layouts to maximize customer satisfaction and sales opportunities (Newcomb, 2018). (Poorni Sakrabani, Ai Ping Teoh, Azlan Amran 2019) Retail 4.0 will enable retailers to create transformative shopping experiences, better inventory management, increased operational efficiency, and more informed real-time decision making. We are now able to overcome these problems. (Youngkeun Choi, 2020) AI-based technology strengthens the relationship between users' ability and willingness to accept AI technology. (Loske, et al, 2021) AI systems have proven to be the most efficient. Therefore, AI capabilities enable systems to achieve specific goals (Haenlein & Kaplan, 2019). Specifically, this ability refers to the ability to simulate human intelligence, especially those involving cognition such as learning and problem-solving, in ever-changing environments based on continuous data collection (Humerick, 2018). (Sohn et al., 2020). Artificial Intelligence (AI) has emerged as one of the biggest disruptors in the consumer market (Hackl & Wolfe, 2017). Unbeknownst to consumers, it is widely applied to various services and products (Krogue et al., 2017). Fashion-conscious and insightful about fashion trends (Bakewell & Mitchell, 2003); (Valaei & Nikhashemi, 2017). Increased knowledge of product features, novelty, and differentiation has been shown to have a positive impact on consumer purchasing behavior (Tanner & Wolfing Kast, 2003). New technological advances and frequent and rapid changes in corporate organizational structures force us to take a new perspective on human capital management based on

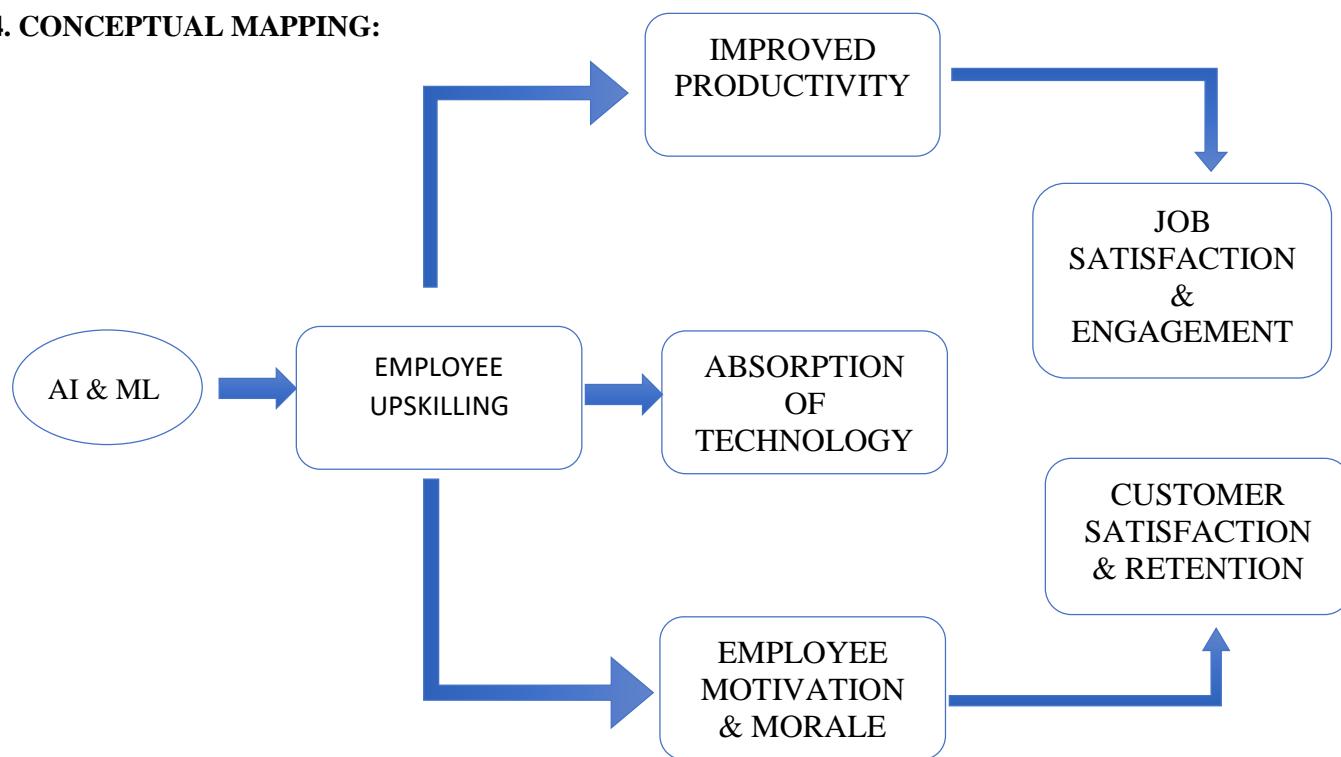


knowledge and collected data. The first step in this direction begins with the introduction of employee monitoring systems in many companies. Monitoring is any system or process used to collect, store and analyze data from multiple sources and report on employee activity and performance (Ball et al., 2010). Use real-time face-to-face communication. HR professionals typically evaluate employee performance on an annual, semi-annual, or perhaps quarterly basis, but this adoption will allow these evaluations to be conducted more regularly, increasing efficiency (Nishad 2019). In the context of consumer use of AI tools, motivation is defined as the behavioral factors that guide a consumer to use her AI (Jin & Kim, 2003). Chatbots, voice assistants, and augmented reality are the most common AI tools consumers use to make purchasing decisions (Turban et al., 2017). Interaction-based technologies are easy to use, help provide information quickly, and reduce human effort (Brandtzaeg & Følstad, 2017). It also helps improve the user experience (Chen & Tsai, 2012). User experience is enhanced by evoking emotions, which positively influence behavioral intentions such as product purchase intentions (Lecointre-Erickson et al., 2018).

### 3. OBJECTIVES:

- To study the Impact of Artificial Intelligence on Retail stores
- To analyze the Impact of Artificial Intelligence on Retail store profitability
- To study the Impact of employee upskilling through Artificial Intelligence

### 4. CONCEPTUAL MAPPING:



### 5. RESEARCH METHODOLOGY:

The qualitative methodology used included an in-depth literature review and synthesizing observations and results made in field studies. Researchers have found direct links to various academic conversations surrounding this area of research. We did a study on the top 5 retail stores using AI in their retail operations. This study was conducted in Chennai, India.

### 6. DATA COLLECTED:

CRITERIA	Implemented tool	Revenue (Current year)	No of Stores	No of Employee
<u>COMPANY</u>				
Future Group	MoEngage Inc's	\$4.6 billion	1,500+	50,000+
Pantaloons	Algonomy	26 billion	344+	25,000+
Bata	Agrex.ai	8.2975 billion	1,375+	30,000+



Arvind Fashions	Nucleus Vision LLC	12.0128 billion	1,300+	25,620+
Reliance	hyperlocal	676.34 billion	14,412+	1,00,000+
Lenskart	Tango Eye	64.374 billion	1,100+	5,000+

Sources: Fashion network, financial express, pantaloons, business standards

## 6.1 INTERPRETATION:

From the above collected data we have studied and analyzed the sales, revenue and number of outlets. This helped us get a deeper knowledge about the stores and helped us with the research.

## 7. STATEMENT OF PROBLEM:

The application of artificial intelligence (AI) in retail operations has the potential to significantly impact profitability. With the rise of e-commerce and online shopping, traditional brick-and-mortar stores face intense competition. Retail stores have to find ways to differentiate themselves and provide unique customer experiences to remain relevant. Consumer shopping habits are constantly evolving, and retailers need to adapt to keep pace. This may involve investing in new technology like AI, revamping store layouts, or adjusting product offerings, these can be made effective using the AI tools available. The retail industry is undergoing a digital transformation, and retailers need to keep up with the latest technology and trends to remain competitive. This includes implementing Omni channel strategies, incorporating artificial intelligence, and improving data analytics. Ensuring a consistent and reliable supply of products can be a challenge for retailers, especially when dealing with unexpected spikes in demand or supply chain disruptions. Retail stores have to balance the need to invest in new technologies and initiatives with the need to keep costs under control and maintain profitability. Overall, the implementation of AI in retail operations and profitability is moderated by the motivation and upskilling of employees. Retailers need to address these challenges to fully leverage the potential of AI to improve their operations and increase their profitability.

## 8. LIMITATION:

- The quality and availability of data may be limited, making it difficult to accurately measure the impact of AI on retail operations and profitability. For example, data on employee motivation and upskilling may be difficult to obtain, or may not be available in a usable format.
- The implementation of AI technology in retail operations may be a slow process, and it may take time to fully realize the benefits of the technology. This means that a study of the impact of AI may need to be conducted over an extended period to accurately capture the benefits of the technology.
- The impact of AI on retail operations and profitability may be complex and may be influenced by several factors, including employee motivation, upskilling, organizational culture, and market conditions. This makes it difficult to isolate and measure the impact of AI.
- Despite the potential benefits of AI, there may be resistance to the adoption of the technology among employees, customers, and other stakeholders. This resistance may limit the success of the technology, and make it difficult to accurately measure its impact.

## 9. FUTURE STUDY:

Further research is needed to understand the impact of AI on employee motivation and job satisfaction, and to identify strategies to mitigate any negative impacts and enhance positive outcomes. A study could be made to examine the effectiveness of different employee upskilling programs and the factors that contribute to their success or failure. Studies could be conducted to better understand the impact of AI on retail operations and profitability, and to identify best practices for leveraging AI to improve performance. Future studies could explore the most effective ways to integrate AI with existing retail systems, including the challenges that need to be overcome and the benefits that can be realized. Further research is needed to examine the ethical implications of AI in retail, including the potential for biases and discrimination, and to identify best practices for ensuring that AI is used responsibly and ethically. Studies could be conducted to understand the impact of AI on the retail supply chain, including the effects on inventory management, procurement, and logistics.

## 10. MANAGERIAL IMPLICATIONS:

The implementation of artificial intelligence (AI) in retail operations and profitability is moderated by employee motivation and upskilling. To ensure a successful implementation, retailers need to consider Employee engagement in which Retail managers need to actively engage with employees to understand their concerns and address any fears they



may have about the implementation of AI. This can be done through open and honest communication, as well as by providing opportunities for employees to learn about AI and its potential benefits. Employee upskilling is the main criterion which Retail managers need to ensure that their employees have the skills and knowledge needed to effectively use and manage AI. This can be done through employee training programs, workshops, and other upskilling initiatives. Integration of existing systems in Retail managers need to ensure that AI is integrated with existing systems seamlessly and effectively. This requires careful planning and consideration of compatibility and data integration. Monitoring performance of employees, Retail managers need to monitor the performance of AI to ensure that it is having the desired impact on retail operations and profitability. Ethical considerations need to be implicated in AI and ensure that it is used responsibly and ethically. This includes avoiding biases and discrimination and ensuring that AI models are transparent and accountable. Retail managers need to continuously evaluate and improve the implementation of AI in their operations. This requires ongoing monitoring and evaluation, as well as regular updates to AI algorithms to ensure that they are up-to-date and effective. By considering these managerial implications, retail managers can optimize the implementation of AI and ensure that it has a positive impact on retail operations and profitability, while also addressing employee motivation and upskilling.

## 11. FINDINGS:

AI will have a positive impact on retail operations. AI can improve efficiency and productivity in retail operations, leading to cost savings, better inventory management, and improved customer experience. The use of AI in retail can lead to improved margins and increased profits, as the technology can help retailers to better understand customer needs and preferences, and optimize pricing and promotions. Employee motivation and upskilling can play a critical role in the impact of AI on retail operations and profitability. For example, motivated employees who are trained in the use of AI technology are more likely to adopt and effectively use the technology, leading to improved results. A supportive organizational culture can help to mitigate resistance to AI adoption and promote employee motivation and upskilling. This can include clear communication about the benefits of the technology, opportunities for employee training and development, and support for employees as they learn to use the technology.

## 12. CONCLUSION:

AI has the potential to have a significant positive impact on retail operations and profitability, providing retailers with the tools they need to improve efficiency, increase sales, and better understand customer needs and preferences. Employee motivation and upskilling play a critical role in the successful adoption and implementation of AI in retail. Retailers who prioritize employee training and development, and who create a supportive organizational culture that encourages the adoption of new technology, are more likely to see positive results from their investment in AI. To fully realize the potential benefits of AI in retail, it will be important for retailers to invest in the technology, as well as in employee training and development programs that will support its successful adoption and implementation.

## REFERENCES:

1. Business Standard, blending tech with brick and mortar: Future Group's formula to grow sales. [https://www.business-standard.com/article/companies/blending-tech-with-brick-and-mortar-future-group-s-formula-to-grow-sales-118122300422\\_1.html](https://www.business-standard.com/article/companies/blending-tech-with-brick-and-mortar-future-group-s-formula-to-grow-sales-118122300422_1.html)
2. Bhatt, M. and Shah, P. (2023), "Acceptance of Artificial Intelligence in Human Resource Practices by Employees", Tyagi, P., Chilamkurti, N., Grima, S., Sood, K. and Balusamy, B. (Ed.) The Adoption and Effect of Artificial Intelligence on Human Resources Management, Part B (Emerald Studies in Finance, Insurance, and Risk Management), Emerald Publishing Limited, Bingley, pp. 13-30. <https://doi.org/10.1108/978-1-80455-662-720230002>
3. Cui, Y.(G.), van Esch, P. and Jain, S.P. (2022), "Just walk out: the effect of AI-enabled checkouts", European Journal of Marketing, Vol. 56 No. 6, pp. 1650-1683. <https://doi.org/10.1108/EJM-02-2020-0122>
4. Chopra, K. (2019), "Indian shopper motivation to use artificial intelligence: Generating Vroom's expectancy theory of motivation using grounded theory approach", International Journal of Retail & Distribution Management, Vol. 47 No. 3, pp. 331-347. <https://doi.org/10.1108/IJRD-11-2018-0251>
5. Choi, Y. (2021), "A study of employee acceptance of artificial intelligence technology", European Journal of Management and Business Economics, Vol. 30 No. 3, pp. 318-330. <https://doi.org/10.1108/EJMBE-06-2020-0158>
6. Cao, L. (2021), "Artificial intelligence in retail: applications and value creation logics", International Journal of Retail & Distribution Management, Vol. 49 No. 7, pp. 958-976. <https://doi.org/10.1108/IJRD-09-2020-0350>
7. Chuang, S. (2022), "Indispensable skills for human employees in the age of robots and AI", European Journal of Training and Development, Vol. ahead-of-print No. ahead-of-print. <https://doi.org/10.1108/EJTD-06-2022-0062>



8. Chaudhuri, S., Krishnan, L.R.K. & Poorani, S. (2022). Impact of using ai in manufacturing industries. *Journal of the International Academy for Case Studies*, 28(S4), 1-10
9. Fashion network, Future group eyes \$3.5 billion revenue from fashion next fiscal. <https://in.fashionnetwork.com/news/future-group-eyes-3-5-billion-revenue-from-fashion-next-fiscal,899328.html>
10. Financial express, Reliance Retail contributed over 63 pc of sales of Future Consumer in FY22. <https://www.financialexpress.com/brandwagon/span-stylefont-family-arial-font-size-13-3333px-font-weight-400-white-space-normalreliance-retail-contributed-over-63-pc-of-sales-of-future-consumer-in-fy22span/2654696/>
11. Heins, C. (2022), "Artificial intelligence in retail – a systematic literature review", *Foresight*, Vol. ahead-of-print No. ahead-of-print. <https://doi.org/10.1108/FS-10-2021-0210>
12. Hammer, A. and Karmakar, S. (2021), "Automation, AI and the Future of Work in India", *Employee Relations*, Vol. 43 No. 6, pp. 1327-1341. <https://doi.org/10.1108/ER-12-2019-0452>
13. Loske, D. and Klumpp, M. (2021), "Intelligent and efficient? An empirical analysis of human–AI collaboration for truck drivers in retail logistics", *The International Journal of Logistics Management*, Vol. 32 No. 4, pp. 1356-1383. <https://doi.org/10.1108/IJLM-03-2020-0149>
14. Mukherjee, I., & Krishnan, L.R.K. (2022). Impact of AI on aiding employee recruitment and selection process. *Journal of the International Academy for Case Studies*, 28(S2), 1-15
15. Maity, S. (2019), "Identifying opportunities for artificial intelligence in the evolution of training and development practices", *Journal of Management Development*, Vol. 38 No. 8, pp. 651-663. <https://doi.org/10.1108/JMD-03-2019-0069>
16. Pradhan, I.P. and Saxena, P. (2023), "Reskilling Workforce for the Artificial Intelligence Age: Challenges and the Way Forward", Tyagi, P., Chilamkurti, N., Grima, S., Sood, K. and Balusamy, B. (Ed.) *The Adoption and Effect of Artificial Intelligence on Human Resources Management, Part B (Emerald Studies in Finance, Insurance, and Risk Management)*, Emerald Publishing Limited, Bingley, pp. 181-197. <https://doi.org/10.1108/978-1-80455-662-720230011>
17. Pantaloons, Who we are? <https://www.pantaloons.com/content/about-us-4>
18. Qamar, Y., Agrawal, R.K., Samad, T.A. and Chiappetta Jabbour, C.J. (2021), "When technology meets people: the interplay of artificial intelligence and human resource management", *Journal of Enterprise Information Management*, Vol. 34 No. 5, pp. 1339-1370. <https://doi.org/10.1108/JEIM-11-2020-0436>
19. Rana, J., Gaur, L., Singh, G., Awan, U. and Rasheed, M.I. (2022), "Reinforcing customer journey through artificial intelligence: a review and research agenda", *International Journal of Emerging Markets*, Vol. 17 No. 7, pp. 1738-1758. <https://doi.org/10.1108/IJOEM-08-2021-1214>
20. Sohn, K., Sung, C.E., Koo, G. and Kwon, O. (2021), "Artificial intelligence in the fashion industry: consumer responses to generative adversarial network (GAN) technology", *International Journal of Retail & Distribution Management*, Vol. 49 No. 1, pp. 61-80. <https://doi.org/10.1108/IJRDM-03-2020-0091>
21. Tschang, F. T., & Mezquita, E. A. (2020). Artificial intelligence as augmenting automation: Implications for employment. *Academy of Management Perspectives*. <https://doi.org/10.5465/amp.2019.0062>
22. Wamba-Taguimdje, S.-L., Fosso Wamba, S., Kala Kamdjoug, J.R. and Tchatchouang Wanko, C.E. (2020), "Influence of artificial intelligence (AI) on firm performance: the business value of AI-based transformation projects", *Business Process Management Journal*, Vol. 26 No. 7, pp. 1893-1924. <https://doi.org/10.1108/BPMJ-10-2019-0411>
23. Weber, F.D. and Schütte, R. (2019), "State-of-the-art and adoption of artificial intelligence in retailing", *Digital Policy, Regulation and Governance*, Vol. 21 No. 3, pp. 264-279. <https://doi.org/10.1108/DPRG-09-2018-0050> (2021), "AI improvements: Adopting AI in the retail sector to gain competitive advantage", *Strategic Direction*, Vol. 37 No. 6, pp. 17-19. <https://doi.org/10.1108/SD-04-2021-0039>

PAPER • OPEN ACCESS

## Intelligent Chatbot Adapted from Question and Answer System Using RNN-LSTM Model

To cite this article: P Anki *et al* 2021 *J. Phys.: Conf. Ser.* **1844** 012001

View the [article online](#) for updates and enhancements.



The Electrochemical Society  
Advancing solid state & electrochemical science & technology

**240th ECS Meeting** ORLANDO, FL

Orange County Convention Center Oct 10-14, 2021



Abstract submission deadline extended: April 23rd

SUBMIT NOW

# Intelligent Chatbot Adapted from Question and Answer System Using RNN-LSTM Model

P Anki<sup>1\*</sup>, A Bustamam<sup>1</sup>, H S Al-Ash<sup>1</sup> and D Sarwinda<sup>1</sup>

<sup>1</sup> Department of Mathematics, Universitas Indonesia Depok 16424, Indonesia

Email: prasnurzaki.anki@sci.ui.ac.id\*

**Abstract.** In modern times, the chatbot is implemented to store data collected through a question and answer system, which can be applied in the Python program. The data to be used in this program is the Cornell Movie Dialog Corpus which is a dataset containing a corpus which contains a large collection of metadata-rich fictional conversations extracted from film scripts. The application of chatbot in the Python program can use various models, the one specifically used in this program is the LSTM. The output results from the chatbot program with the application of the LSTM model are in the form of accuracy, as well as a data set that matches the information that the user enters in the chatbot dialog box input. The choice of models that can be applied is based on data that can affect program performance, with the aim of the program which can determine the high or low level of accuracy that will be generated from the results obtained through a program, which can be a major factor in determining the selected model. Based on the application of the LSTM model into the chatbot, it can be concluded that with all program test results consisting of a variety of different parameter pairs, it is stated that Parameter Pair 1 (size\_layer 512, num\_layers 2, embedded\_size 256, learning\_rate 0.001, batch\_size 32, epoch 20) from File 3 is the LSTM Chatbot with the avg accuracy value of 0.994869 which uses the LSTM model is the best parameter pair.

## 1. Introduction

Issues submitted by consumers, in general, can be accessed through a number of questions that have been through a question and answer system, which can relate to various storage in data, limited customer service that cannot function fully for a whole day, so the program is needed to work optimally and produce service results in the form of answers to questions raised by consumers. Choosing chatbots as a solution for answers of questions based on various problems that consumers issue can make it easier for consumers to get answers. In arranging the structure of the display components on the chatbot into the Python program, because it is easy to use, and more productive in interpretation programs [1]. Based on set of sequences, computational methods must solve two major problems: effectively representing a sequence as a feature vector that can be analyzed and designing a model that can identify data accurately and quickly [2]. In the end, it is hoped that the input data generated through the question and answer system will be prepared, which will then be implemented into the Python program so that in the end, the consumers can get answers to the questions they raised to the chatbot. Chatbot is a system that accepts user input with an ongoing response, parts of the chatbot can be built from an encoder-decoder architecture [3]. Simply put, a chatbot is a simple robot in the form of a program to answer questions from users, which produces output data in the form of answers. In spite of users' low satisfaction and continuance intention (CI) regarding chatbots, few studies have explored why consumers are reluctant to continue using them. In light of this, empirical investigation of the users' satisfaction with chatbots and its CI becomes relevant [4]. When the research results can understand the information needs needed



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

by consumers with the linkage of information systems in the chatbot, it is expected that consumers' satisfaction will also increase.

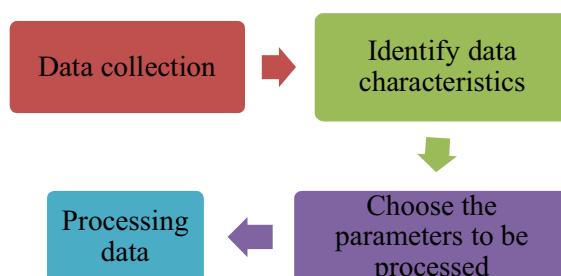
There is a high degree of variance in chatbot quality. Our studies compare a hypothetically perfect chatbot (error-free) to a chatbot which struggles to infer meaning and thus seeks clarification (clarification) to a third chatbot which produces an error in comprehension (error). Predicting that the error-free chatbot will outperform the error producing chatbot is straightforward [5]. In this research will compare how accurately the chatbot can meet the information needs required by the user, so it can also be seen how much error is caused due to the mismatch of the information required. In section 2, we will discuss about materials and method will be used in this research, among others are steps in making chatbot, which will tell the reader what steps are required to create a chatbot, discussion regarding LSTM models, which will provide info regarding what the LSTM model is and how it is implemented, displays a study of the greedy method, which will help improve program performance, conduct lessons related to the seq2seq model, to help translate sentences in the program.

## 2. Materials and method

There are set of sentences from dataset, which can used to build chatbot program based on LSTM model, multiple parameter pairs, Greedy method. Input from the user can be command to run chatbot program, with the result that set of sentences which contain information according to the input that the user enters.

### 2.1. Steps in making a chatbot

There are 4 steps to creating a chatbot, namely: data identity, data input for the question and answer system, compiling a chatbot program, and evaluating the output.



**Figure 1.** Data management diagram

- Data identity

The data to be used in this program is the Cornell Movie Dialog Corpus which is a dataset containing a corpus which contains a large collection of metadata-rich fictional conversations extracted from film scripts, which have 220,579 conversation exchanges between 10,292 pairs of movie characters, involving 9,035 characters from 617 films, and a total of 304,713 sayings [6]. The data used is the 2018 data. Based on the parameter selection stage, the conversation data will then be selected and then processed from the input questions that come from the dataset, which will then be answered by the machine as a form of response to the chatbot program. The answers to the questions will produce a response for individuals who want to obtain answers about matters related to the characteristics of film data. To manage this data, below is a data management diagram that explains the steps from collecting to processing the data.

- Question and answer system input data

In carrying out input from program users, files containing dialog sentences in the film are inputted into the program as user input. Then, the input will be processed to obtain the output of the program in the form of a dialogue sentence in the film, by having a relationship between the dialog in the input to the dialog in the output.

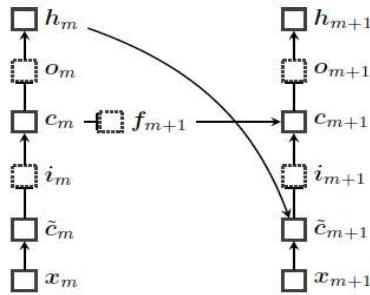
- Chatbot program development

The following are the things that need to be considered in the preparation of the chatbot program: there will be various translation choices in the form of sequence-to-sequence, and the selection of the LSTM model so that we can determine the best accuracy from the response of the chatbot compared to the live human response.

- Output evaluation

The last step is evaluating whether or not the model has provided accurate results. For example, there may be a relationship between the input dialog and the output dialog that turns out to be much more accurate than an input dialog selection that the program user previously entered.

## 2.2. LSTM Model



**Figure 2.** LSTM architecture [3]

The Long Short-Term Memory (LSTM) model is a special type of RNN that has the ability to study long-term dependencies [7]. In this model, the encoder is applied to the last hidden statement of the LSTM [3]. The LSTM architecture is shown in Figure 2, and the full equation for updates is as follows from [3]:

$$f_{m+1} = \sigma(\theta^{(h \rightarrow f)} h_m + \theta^{(x \rightarrow f)} x_{m+1} + b_f) \text{forget gate} \quad (1)$$

$$i_{m+1} = \sigma(\theta^{(h \rightarrow i)} h_m + \theta^{(x \rightarrow i)} x_{m+1} + b_i) \text{Input gate} \quad (2)$$

$$\tilde{c}_{m+1} = \tanh(\theta^{(h \rightarrow c)} h_m + \theta^{(w \rightarrow c)} x_{m+1}) \text{update candidate} \quad (3)$$

$$c_{m+1} = f_{m+1} \odot c_m + i_{m+1} \odot \tilde{c}_{m+1} \text{memory cell update} \quad (4)$$

$$o_{m+1} = \sigma(\theta^{(h \rightarrow o)} h_m + \theta^{(x \rightarrow o)} x_{m+1} + b_o) \text{Output gate} \quad (5)$$

$$h_{m+1} = o_{m+1} \odot \tanh(c_{m+1}) \text{Output} \quad (6)$$

The operator  $\odot$  is an elementwise product. The LSTM model is the result of adding a hidden state which is expressed in  $h_m$  with the memory cells represented as  $c_m$ . The value of the memory cell at each  $m$ th time is the gate form which is a sum of two quantities: the value of the previous memory cell expressed with  $c_{m-1}$ , and the value of the memory cell that has undergone an update expressed with  $c_m$ , which is calculated from the previous input in the current  $x_m$  form and the previous hidden state in the  $h_{m-1}$  form. The next state is  $h_m$  calculated from the cell memory. Since memory cells do not pass through non-linear function pathways during renewal, it is possible for information over remote networks [3].

Each gate is controlled by a vector that has a weight, which determines the previous hidden state (e.g.,  $\theta^{(h \rightarrow f)}$ ) and Input current (e.g.,  $\theta^{(x \rightarrow f)}$ ), plus the vector offset (e.g.,  $b_f$ ). The overall operation can be informally summarized as  $(h_m, c_m) = \text{LSTM}(x_m, (h_{m-1}, c_{m-1}))$ , with  $(h_m, c_m)$  representing the LSTM status after reading the  $m$  token. LSTM outperforms the standard artificial neural network on

a wide range of issues. The gates are shown in squares with dotted edges. In the LSTM language model, each  $h_m$  will be used to predict the next word  $w_{m+1}$ . LSTM outperforms standard recurrent neural networks in a variety of problems, for example in language modeling problems [3].

Based on the discussion of [3], Recurrent neural networks (RNNs) were introduced as a language modeling technique, where the context in token m (where the token stops) as well as token M (the specified M token value) is summarized by the repeated updated vector,

$$h_m = g(x_m, h_{m-1}), m = 1, 2, \dots, M, \quad (7)$$

where  $x_m$  is the stored vector of the  $w_m$  token and the function  $g$  defines the loop. The initial condition  $h_0$  is an additional parameter of the model. The LSTM (Long Short Term Memory) model has a more complex loop, in which the memory cells pass through a series of gates, avoiding repeated applications of non-linearity. A Gate is an input function and is the previous hidden status. The form is calculated from the activation of the elementwise sigmoid,  $\sigma(x) = (1 + \exp(-x))^{-1}$ , ensuring that the value will be in the range of [0, 1]. Therefore, this form can be seen as a differentiated logic gate.

The LSTM model is one of many parallel computing models. Based on reference from [8], parallel computing can be used to process the program instructions by distributing them into multiple processors in order to reduce the running time of the program.

### 2.3. Greedy method

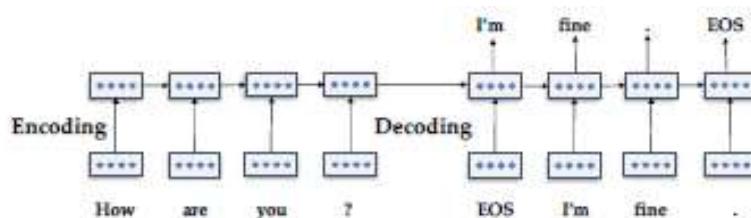
In addition to choosing the model, the next step is to determine the method used in the program. In this program, the greedy method is chosen as a form of implementing the LSTM model so that when running the program, the data processing time can be faster while also increasing the accuracy of the selected model [9]. Based on the discussion of the greedy algorithm in [10], the greedy algorithm basically chooses the best decisions by looking at the closest estimate of each stage of the problem in the hopes of finding a good solution. In this case, the algorithm selects the lowest-cost worker who is then assigned to the task as the first task, then selects the next lowest-cost worker to be assigned to that task, and so on until all the tasks have been assigned. The following is a case example of applying the greedy algorithm for the Linear Assignment Problem (LAP):

- Step 1: Find the location of the smallest element, and delete the location from the rows and columns.
- Step 2: Find the location of the second smallest element, and delete the locations of the rows and columns.
- Step 3: Repeat the process until there are no rows and columns to delete.
- Step 4: Select the given result as the optimal local result.

The greedy algorithm tries to approach the optimal solution by increasing the candidate solution iteratively, without any guarantee that the optimal solution will actually be found. Greedy's algorithm itself will find good answers, but it can scramble parts of the algorithm, but will ensure that many answers are possible. Modifying the greedy algorithm so that it sometimes accepts assignments at certain periods of time which can worsen the objective function, but will then succeed in obtaining local optimizations and possibly finding optimal global solutions.

### 2.4. Seq2seq model

Things that need to be known regarding the importance of implementing the seq2seq model are as follows:



**Figure 3.** Decoder Encoder Model for Neural Response Generation in Dialogue [9]

For the software used, Jupyter Notebook software based on Python is chosen so that program users can clearly see the input and output of the program being run. When implementing a question and answer system into a program, it is necessary to have a seq2seq model, which can function to produce various responses to user input [9]. Things that need to be known, regarding the importance of implementing the seq2seq model are as follows:

- In its application as the basis of the encoders model, an entered input has a tendency to produce a predictable but repetitive response to close the conversation.
- The problem above can be overcome by changing the objective function for training the seq2seq model into mutual information destinations or by modifying the decoder to make it more diverse in response to the things that are given.

In section 3, we will discuss about data implementation in the python program, which will explain what is needed to create a chatbot program using the python program.

### **3. Data implementation in the python program**

To implement the data into the Python program with the Jupyter Notebook software, a program planning plan will be systematically compiled as follows:

a. Choosing the software to be used

In choosing software, the main things to be considered are things such as whether the selected data will be processed properly in the program, the performance of the software in processing data, and the availability of supporting attributes for programmers needed in making the program.

b. Selecting supporting models and attributes

Choosing a model that is in accordance with the characteristics of the data can affect program performance. Determining a high or low level of accuracy that is generated from a program is a major factor in choosing the model. Based on the consideration of the model selection requirements, the LSTM model was selected as model to be applied into the program. In addition, supporting attributes (such as the seq2seq model), are the next determining factors that can verify whether the data processing process matches the criterias that was meant to guide the process. In application to the program, the seq2seq model can process input sentences which will then be processed with other models and structures in the program, so that in the end it can issue various output sentences as a response generated from the chatbot program.

c. Determining the program evaluation method

In this case, there are various program evaluation methods, such as loss, accuracy, val\_loss, and val\_accuracy. Based on the case example from [9], to introduce a method for evaluating text classification, some simple binary detection tasks will first be considered.

For example, in spam detection, the aim is to label any text that is in the spam category in the form of a label ("positive") or not in the spam category in the form of another label ("negative"). For each item in the form of an email document, it is necessary to know whether the system created can determine whether an item is spam or not.

What needs to be known to determine whether an email is actually spam or not, for example, the human-defined labels for each document will be tested for suitability. The mention of human labels will then be referred to as the gold table. Before testing spam detection, it is necessary to prepare a standard measurement to find out how well the spam detection has performed. One thing to pay attention to to evaluate any system for detecting something is to start with constructing a Contingency Table as shown in Table 1.

**Table 1.** Contingency Table [9]

		gold standard labels		
		system positive	system negative	precision $= \frac{tp}{tp + fp}$
system output labels	system positive	true positive	false positive	
	system negative	false negative	true negative	
		recall $= \frac{tp}{tp + fn}$		accuracy $= \frac{tp + tn}{tp + fp + tn + fn}$

Each cell is labeled as the set of possible outcomes. In the case of spam detection, for example, a true positive is a document that is spam (shown in the Contingency Table) and the result from the system says it is spam. False negatives are documents that are spam but result from the system labeling them non-spam. True negatives are documents that are spam and result from the system labeling them as spam. False positives are documents that are non-spam but the results from the system labels them as spam. In the lower right corner of the Contingency Table there is the form of the equation for accuracy, as follows:

$$\text{accuracy} = \frac{tp + tn}{tp + fp + tn + fn} \quad (8)$$

The following are matters that need to be prepared to obtain the loss value. Based on the theory of [11], for example, for an observation  $x$ , the loss function is stated as in the equation below:

$$L(\hat{y}, y) = \text{The amount of difference from the true } y \text{ value} \quad (9)$$

In calculation to determine how close the output classifier ( $\hat{y} = \sigma(w \cdot x + b)$ ) is to the actual output ( $y$ , which is 0 or 1).

Based on the definition in [11], validation loss is an estimate of the error rate derived from the model that is calculated based on the loss function of the validation set, which is observed through the training process. After the model has been selected and trained, it is necessary to evaluate how effective this model is for the purposes of the classification task. Intuitively, one way to evaluate a model is to calculate the percentage of samples that have been correctly classified. So, the classification rate can be calculated from the following model:

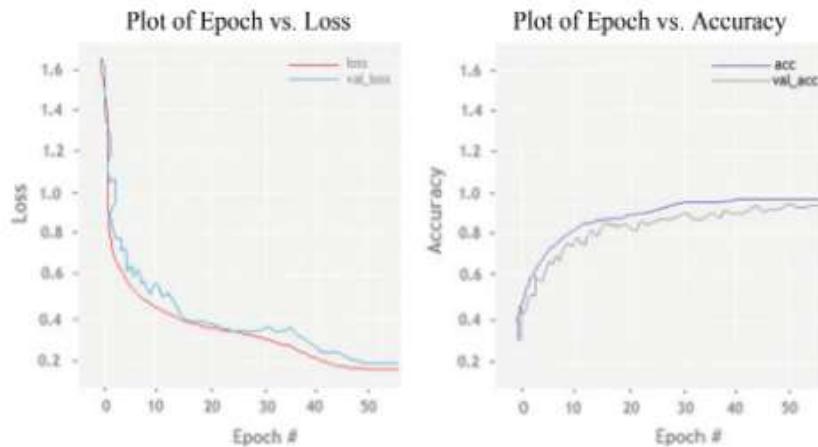
$$\text{classification rate} = \frac{\text{number of samples correctly classified}}{\text{total number of samples}} \quad (10)$$

Based on the equation above, classification errors can be calculated and are complementary to the classification rates. The calculation of the error rate of the model can be calculated as follows:

$$\text{error rate} = \frac{\text{number of loss function over validations set}}{\text{total number of validations set}} \quad (11)$$

The percentage generated by calculating the loss function against the set of validations comes from a predetermined model. Based on the case example from [12], we can see the comparison between the

loss from the training dataset set against the val\_loss from the validation set, through the graph in Figure 4.



**Figure 4.** Comparison chart of epoch to loss and comparison chart of epoch to accuracy [12]

Based on the graph in Figure 4, it can be seen that the validation loss (val\_loss) can be lower or higher than the loss value of the training dataset. In other words, it can be called underfitted or overfitted. Based on the case example from [12], we can see the comparison between the accuracy of the training dataset (acc) to the validation accuracy (val\_acc) of the validation set, through the graph in Figure 4. Because the training dataset is data that is known from the model and validation data is data that is unknown from the model, the acc value is generally higher than val\_acc. If the validation accuracy value (val\_acc) is lower or higher than the accuracy value of the training dataset, it can be said to be underfitted or overfitted. After deliberating the comparison of the elaboration of each program evaluation method, accuracy is chosen as the appropriate program evaluation method for the chatbot program.

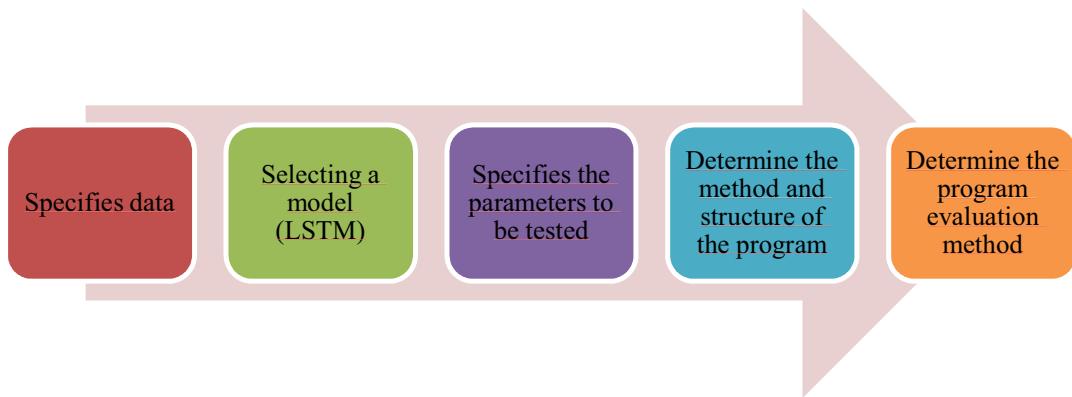
In section 4, we will discuss about applying data implementation in the python program, which will be more detail with description of the problem which will explain how the program description should be created, and program making which will inform the reader how to make chatbot based on flow of programming that has been planned.

#### 4. Applying data implementation in the python program

This elaboration will explain the description of the problem that can be used as a study of the need for implementing chatbots in the question and answer system.

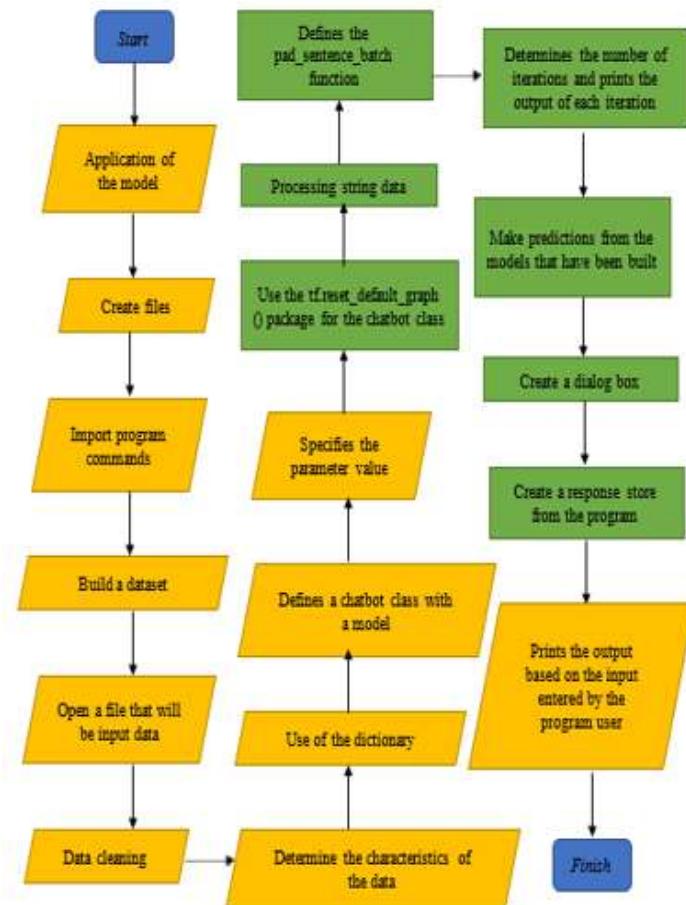
##### 4.1. Description of the problem

In facing the dynamics of modern times, there will indeed be numerous questions that consumers want to ask, based on things related to a specific data. Based on the speed comparison between manual question and answer service system carried between humans with the resulting response based on the chatbot program, which is the implementation of question and answer data between humans and machines, as well as easy access and operation times according to needs of the chatbot program user, these are the reasons why it is necessary to create a chatbot program, as a form of implementation of the question and answer system data. Before discussing the components needed to make a chatbot, it is necessary to know the things that must be prepared to obtain the components needed for making the program as shown in the diagram below:



**Figure 5.** Preparation diagram of the components for the chatbot program

#### 4.2. Program making

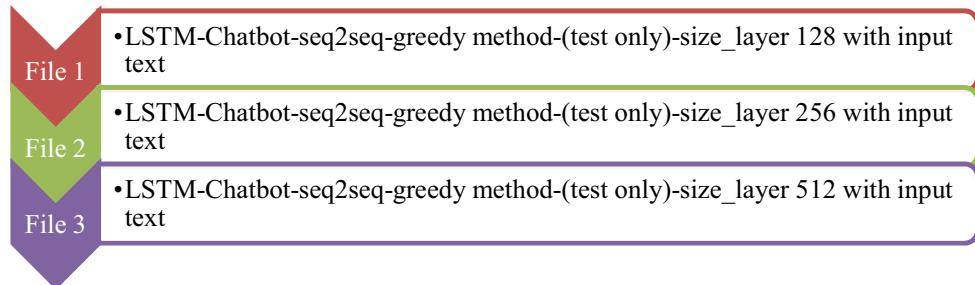


**Figure 6.** Program making flowchart

In section 5, we will discuss about program creation detail and discussion of test results, in this section will discuss the description of the file that will be named according to the parameter pair used, as well as the test results that have been tested in the program.

## 5. Program creation detail and discussion of test results

After the chatbot program has been created, there will be 3 files generated as a result of implementing the LSTM model into the chatbot program, as shown below:

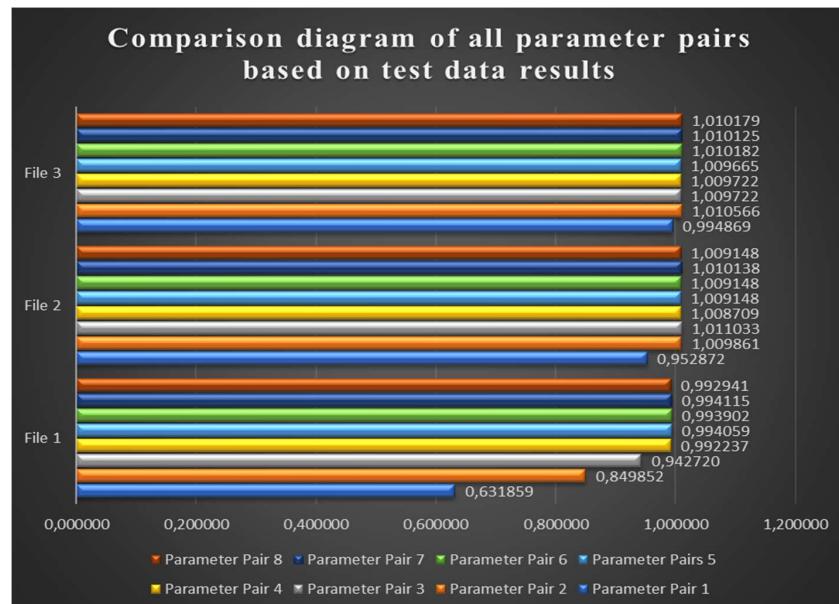


**Figure 7.** Detail 3 chatbot program files

**Table 2.** Data to be Tested in LSTM Chatbot

Parameter Pair	File 1	File 2	File 3
<i>size_layer</i>	128	256	512
<i>num_layers</i>	2	2	2
<i>embedded_size</i>	64	128	256
<i>learning_rate</i>	0.001, 0.0015	0.001, 0.0015	0.001, 0.0015
<i>batch_size</i>	8	16	32
<i>epoch</i>	20,30,40,50	20,30,40,50	20,30,40,50

In Table 2, it can see the various parameter pairs that will be tested in the LSTM chatbot program. Furthermore, with a total of 24 different parameter pairs (different numbers are on size\_layer, learning\_rate, epoch), details of the 8 parameter pairs in 3 different files will be presented with the results of all tested parameter pairs are shown in Figure 8.



**Figure 8.** Comparison diagram of all parameter pairs based on test data results

**Table 3.** Best Test Result Data from Each File

Parameter Type	File 1 <i>Parameter Pair 7</i>	File 2 <i>Parameter Pair 1</i>	File 3 <i>Parameter Pair 1</i>
<i>size_layer</i>	128	256	512
<i>num_layers</i>	2	2	2
<i>embedded_size</i>	64	128	256
<i>learning_rate</i>	0.001	0.001	0.001
<i>batch_size</i>	8	16	32
<i>epoch</i>	50	20	20
<i>avg accuracy</i>	0.994115	0.952872	0.994869

Based on the parameter pairs in each of the following files, the best parameter pair from the 3 files that have been tested will not be assessed if  $> 1.0$  an overfit condition occurs (when the training accuracy results are very good but the testing accuracy results are not as good). What will be selected is the one that produces the best avg accuracy on a scale of 0.0 to 1.0 in the Table 3. So, based on all the test results of the program that has been carried out, it can be stated that the Parameter Pair 1 originating from File 3 is the LSTM Chatbot with the avg accuracy value of 0.994869 is the best parameter pair. The resulting accuracy in the chatbot research using English using the LSTM model is 70.9% [13]. When viewed from these references, in this study it can be stated that there has been an increase in accuracy compared to previous studies.

In section 6, we will discuss about conclusion and future work of this research, which is conclusion of the results in this research, and future work which is relating to things that need to be improved from this research.

## 6. Conclusion and future work

Based on the application of the LSTM model into the chatbot, it can be concluded that with all program test results consisting of a variety of different parameter pairs, it is stated that Parameter Pair 1 (*size\_layer* 512, *num\_layers* 2, *embedded\_size* 256, *learning\_rate* 0.001, *batch\_size* 32, *epoch* 20) from File 3 is the LSTM Chatbot with the avg accuracy value of 0.994869 which uses the LSTM model is the best parameter pair. For future works, we can try improving the LSTM chatbot algorithm using advanced computing environment [15].

In the future work, it is expected that there will be an increase in accuracy from the updating of methods, models, and references that are increasingly following technological developments.

## 7. References

- [1] Raj S 2018 *Building Chatbots with Python: Using Natural Language Processing and Machine Learning* (New York City: Apress) p. 33
- [2] Bustamam A, Musti M I S, Hartomo S, Aprilia S, Tampubolon P P and Lestari D 2019 Performance of rotation forest ensemble classifier and feature extractor in predicting protein interactions using amino acid sequences *BMC Genomics* vol 20 (London: BMC Genomic) p. 2
- [3] Eisenstein J 2018 *Natural Language Processing* (Cambridge: MIT press) pp. 137-138, p. 345
- [4] Ashfaq M, Jiang Y, Shubin Y and Loureiro S M C 2020 I, Chatbot: Modeling the determinants of users' satisfaction and continuance intention of AI-powered service agents *Telematics and Informatics* vol 54 (Amsterdam: Elsevier) p. 2
- [5] Sheehan B, Hyun S J and Gottlieb U 2020 Customer service chatbots: Anthropomorphism and adoption *Journal of Business Research* vol 115 (Amsterdam: Elsevier) p. 15
- [6] Chidananda R 2018 *Cornell Movie-Dialogs Corpus* (Kaggle datasets)
- [7] Muzaffar S and Afshari A 2019 Short-Term Load Forecasts Using LSTM Networks *Energy Procedia* vol 158 (Amsterdam: Elsevier) p. 2922

- [8] Ardanewari G, Bustamam A, Siswantining T 2017 Implementation of parallel k-means algorithm for two-phase method biclustering in Carcinoma tumor gene expression data *AIP Conference Proceedings* vol 1825 (Maryland: AIP Conference Proceedings) p. 2
- [9] Jurafsky D and Martin J H 2019 *Speech and Language Processing An Introduction to Natural Language Processing* (New Jersey: Prentice Hall) p. 66, p. 81, p. 163, p. 497
- [10] Güneri Ö İ, Durmuş B and Aydin D 2019 Different Approaches to Solution of The Assignment Problem Using R Program *Journal of Mathematics and Statistical Science* vol 5 (Delaware: SSPub) p. 134
- [11] Clavance L 2019 *An Evaluation of Machine Learning Approaches to Natural Language Processing for Legal Text Classification* (London: Imperial College London)
- [12] Kotecha K, Piuri V, Shah H N, Patel R 2020 *Data Science and Intelligent Applications: Proceedings of ICDSIA 2020* (New York: Springer Nature) p. 117
- [13] Peters, F 2018 *Master thesis : Design and implementation of a chatbot in the context of customer support* (Belgium: University of Liège) p.47
- [14] Muradi H, Bustamam A and Lestari D 2015 Application of hierarchical clustering ordered partitioning and collapsing hybrid in Ebola Virus phylogenetic analysis *2015 International Conference on Advanced Computer Science and Information Systems* (New York City: IEEE) p. 323

#### Acknowledgments

This research is partially supported by PUTI Prosiding 2020 research grant by DRPM Universitas Indonesia with contract number NKB-927/UN2.RST/HKP.05.00/2020.

# Jewelry Shop Conversational Chatbot

Safa Zaid

Aswah Malik

Fatima Kisa

National University of Computing and Emerging Sciences (ISB)

## Abstract

Since the advent of chatbots in the commercial sector, they have been widely employed in the customer service department. Typically, these commercial chatbots are retrieval-based, so they are unable to respond to queries absent in the provided dataset. On the contrary, generative chatbots try to create the most appropriate response, but are mostly unable to create a smooth flow in the customer-bot dialog. Since the client has few options left for continuing after receiving a response, the dialog becomes short. Through our work, we try to maximize the intelligence of a simple conversational agent so it can answer unseen queries, and generate follow-up questions or remarks.

We have built a chatbot for a jewelry shop that finds the underlying

objective of the customer's query by finding similarity of the input to patterns in the corpus. Our system features an audio input interface for clients, so they may speak to it in natural language. After converting the audio to text, we trained the model to extract the intent of the query, to find an appropriate response and to speak to the client in a natural human voice. To gauge the system's performance, we used performance metrics such as Recall, Precision and F1 score.

**Keywords:** Chatbot, Generative, Natural Language, Performance Measure

## 1 Problem statement

Chatbots are increasingly been used in customer service departments since their introduction into the business

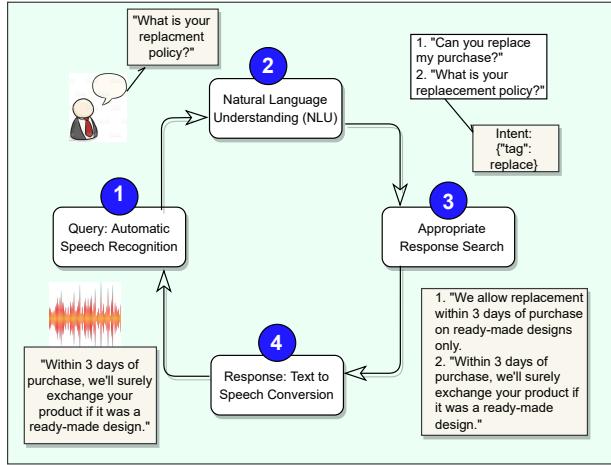


Figure 1: Workflow

sector, especially for online businesses. Unfortunately, basic task-oriented chatbots are unable to address client enquiries that are not present in their Frequently Asked Questions (FAQ) dataset. We will employ state-of-the-art Natural Language Processing (NLP) technology to construct a conversational chatbot that can speak more robustly with its clients. It will be capable of responding to a wider range of queries and presenting the customer with occasional prompts. This will make the customer-chatbot conversation more natural and will promote brand loyalty.

## 2 Introduction

- Problem Details* Jewelry shops are rarely open 24/7, thus their selling time is constrained. For an online jewelry shop, though its website is accessible at any time of the day from across the globe, it is not feasible for the shop to satisfy individual queries of each of its clients. The employment of a chatbot on the jewelry shop's website will allow customers across the globe to enquire about the shop at any time of the day. This better customer service will help retain clients and increase sales. Most chatbots that we interact on websites can answer only a given set of queries since they are rule-based chatbots. This means that if a query does not exactly match a previously saved pattern in the model's corpus, the bot would be unable to respond to it. Using NLP and Machine Learning (ML) models, we developed a conversational chatbot which not only resolves customer issues but also generates follow-up questions and

remarks, making the conversation more human-like for the customer. [14]

The image below shows how a Rule-based and AI-based chatbot are different.

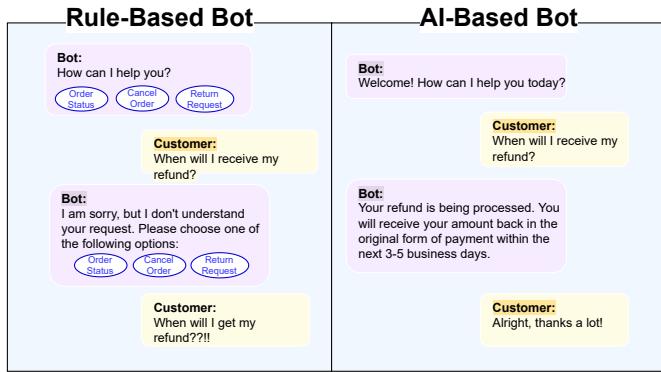


Figure 2: Rule-based vs. AI-based Chatbots

2. *Motivation* Buying and selling items or services through text-messaging applications is part of conversational commerce. Companies are putting a lot of money into digitizing customer service via social media and company websites. They hope to create a more personalized sales experience and stay competitive in the market by these means. A competent sales chatbot will not only respond to consumer questions,

but will also provide product recommendations based on the user's apparent preferences, imitating the work of a salesman in a physical store. Conversational AI, which includes speech recognition, sentiment and semantic analysis, and context-based response creation, is found to assist in creating a personalized customer experience. 52 percent of organizations indicated they increased their usage of automation and conversational interfaces following COVID-19, and 86 percent of respondents said AI has become "mainstream technology" in their company [6].

### 3. *Background*

A dialog system or conversational agent communicates with users in natural language, that is text and/or speech. They can be divided into two classes:

- (a) *Task-oriented dialog agents:* These perform the basic purpose of following the given directions or answering questions on corporate websites.

(b) *Chatbots*: These are designed for extended, unstructured and sometimes even multi-contextual conversations. They can be used both for entertainment and for making the interactions of task-oriented agents more natural.

There are two basic types of chatbots- Rule-based and Artificially Intelligent(AI) or Corpus-based chatbots. The first actual chatbot was rule-based . Rule-based models are simpler to implement but have limited capabilities. They answer queries by pattern matching and thus, can often produce faulty or no solutions when the user query does not match with any recognized pattern. Contrarily, AI models are primarily based on machine learning algorithms which use existing corpora of human conversations to train them. Unlike Rule-based models, AI-based models can understand the user intent and context, and over time, use negative feedback on their mis-

takes to improve performance.

Within AI-based chatbots, there are two further sub-types, namely Information-Retrieval(IR) chatbots and Generative chatbots. Information Retrieval models are trained on a textual dataset, primarily designed to retrieve the information based on user input. The knowledge base for this type of model is usually formed using a database of query-answer pairs. When the person queries the chatbot, the model finds similarities in the query and the chat index.

Generative Models generate entirely new sentences based on the user queries. However, they need to be trained on a large dataset of phrases and real conversations. The model learns sentence structure, syntax, and vocabulary with the aim of generating linguistically correct and contextually appropriate answers.

Neural Networks (NN), first introduced in the late 1980's are large computational networks that are trained on large datasets

in order to approximate some complex target function. They are computational systems that try to solve problems like a human brain, and hence can be used to solve problems like natural language understanding, intent classification and question answering.

### 3 Related work

Digital commerce has resulted in customers demanding round-the-clock customer service by businesses. Due to this, chatbots are increasing in popularity among businesses and consumers alike. More and more companies are ready to pay high amounts of money for the development of these chatbots. As chatbots raise customer engagement via messaging, text, or speech, they are deployed on social and work platforms such as Facebook Messenger, WhatsApp, WeChat, and Slack.

Our chatbot is inspired by many chatbots that we have around ourselves. Early conversational systems like ELIZA [7] (in 1966) and ALICE [8] (in 1995), which were rule-based and had a constrained scope, held

the purpose to mimic human-human text-based conversation. However, the rules were hand-written and responses were generated by keyword pattern matching [9].

In 2000, another major dialog system was introduced, called the DARPA communicator program [10]. Its key features were goal-oriented natural language understanding of requests, conducting dialog and performing tasks. Further, this chatbot had a Learning-based model that used statistical models for understanding spoken inputs in addition to textual inputs. However, its biggest technical limitation was that its performance was poor outside of its well-defined domains.

In 2011, Siri [11], the first widely deployed learning-based Intelligent Personal Assistant (IPA), was developed with an open domain using a Deep Neural Network to convert acoustic patterns in the input voice to form a probability distribution over speech sounds. Like other IPAs, Siri provides both reactive assistance -like generating weather reports- and proactive assistance -like reminding of a friend's birthday- to users so that they could

accomplish a variety of tasks. However, it lacks emotional engagement with its users.

The first widely deployed social chatbot, XiaoIce [12] was designed in 2014 and is used to date. In addition to assisting users in various tasks, it has its own personality and has the ability to create emotional attachment with its users using Emotional Intelligence learning based models in an open-domain using text, speech and images. However, it often shows inconsistent responses and personality traits in long conversations.

In previous years, Sequence-to-Sequence models[13], a special class of RNNs, were used for obtaining valuable results after training on open-domain knowledge. They can also be integrated with other algorithms for domain-specific analyses. Nonetheless, the major drawback of these models is that the entire information (including the past context) of the input sentence into fixed length context vector. Thus, as the sentence or context gets longer, more information is lost and the model responds with decreasing coherence.

## 4 Methodology

Our chatbot has an audio input interface for the customers, meaning the customers can speak to the chatbot in natural language. This audio is converted to text by Python’s Speech Recognition library, SpeechRecognizer [1]. This text is then associated to certain fixed intent in the corpus. Against each intent, we have multiple equivalent responses. Thus, after the customer pattern has been classified as belonging to a particular intent, a seemingly random response is generated. Thereafter, even if the customer asks the same question repeatedly, the response generated is very likely to vary, as well. Furthermore, the chatbot occasionally presents the customer with a followup remark or question to imitate the human conversation. To give our human-chatbot conversation a more natural touch, the chatbot also speaks to the customer in the voice of a man. For this feature, we used the Python library, pyttsx3[2]. The chatbot will continue the conversation with the customer until it classifies an input pattern as a “goodbye” intent.

In the case that a “goodbye” pattern is found, the chatbot greets its client appropriately and ends the conversation.

We developed the chatbot in three different ways:

1. For our first method, we built the chatbot based on TensorFlow’s Keras Sequential model -a feed forward multi-layer neural network. The customer’s input query is pre-processed and compared with the template “patterns” or “queries” in our self-generated customer service dataset. The pre-processing steps include tokenizing, stemming, lemmatization and removing punctuation from our dataset. The input and output layers of the Neural Network consist of One-Hot-Encoded (OHE) embeddings to describe patterns and predicted intents respectively. During the model’s feed forward pass, it optimizes the layer weights using Stochastic Gradient Descent (SGD) and has a standard learning rate of 0.01. The model uses the

Rectified Linear Unit (ReLu) as the activation function between outputs and inputs of adjacent hidden layers. At the last layer, Softmax is applied to our multinomial linear regression model to normalize the output layer results.

2. In the second method, the embeddings from One-Hot-Encoding were replaced by embeddings generated by SentenceTransformer model. This was done to observe how naive One-Hot-Encoded embeddings and the more meaningful SentenceTransformer embeddings of size 384x1 would affect the classifier model’s predictions.
3. In the last method, the SentenceTransformer model from the previous variation was used, but the Intent Classification Model was replaced with a Cosine Similarity Function. This function determined the pattern from the corpus to which the input customer query is most similar. The

intent of the matched pattern is extended to the input and the query is assigned its tag. Finally, a response and optional followup is generated as mentioned above. Following is an example of the final method's working:

Input Query: What time can I visit your shop?

Matched Pattern: What are your shop timings?

Predicted Intent: Timing

Response: Our shop opens at 8 am and closes at 11 pm.

Follow up: We are open for the longest hours in the market!

## 5 Evaluation and Experiments

Our chatbot was built with features many chatbots do not contain. For example, most bots take input and produce output as text, which is not how humans naturally communicate. To avoid a tedious conversation, the chatbot is enabled with the feature

of Speech Recognition. However, a clear voice and quite environment is required for ensuring an appropriate output.

We applied stemming on the user input to easily identify different forms of a word having a similar effect on intent classification. We tested this feature by saying different sentences with different sentence structures but same vocabulary to check if the bot intelligently finds the stem word and responds correctly. For example:

Input Query 1: What time can I visit your shop?

Stemmed Query 1: What time can I visit your shop?

Input Query 2: When is your shop open for visiting?

Stemmed Query 2: When is your shop open for visit ?

Pattern for Queries 1 & 2: What are your shop timings?

Intent for Queries 1 & 2: Timing

An interesting feature of our chatbot is that it does not produce the same response on a repeated query. For this, we have used the ran-

`dom.shuffle()` utility from Python’s random library on the list of responses in our corpus. In addition, it sometimes asks the customer follow up questions for their better understanding unlike other chatbots, which and never initiate the conversation themselves.

First of all, the experiment we conducted was to give same input again and again to confirm that our chatbot always gives a different answer to same input considering that the customer didn’t understand its previous response as demonstrated in example below.

Query 1: What time can I visit your shop?  
Response 1: Our shop opens at 8 am and closes at 11 pm.

Query 2: What time can I visit your shop?  
Response 2: You can come anytime between 8 am and 11 pm!

We evaluated our chatbot using inputs from different intents to calculate its F1 score using a Confusion Matrix for each of the three implementations.

As can be seen from the table above, the One Hot Encoding generated naive and somewhat meaningless

Implementation	F1 Score
OHE with NN	0.592
Sentence Embedding with NN	0.649
Sentence Embedding with Cosine Similarity	0.852

embeddings for the Neural Network classifier. Further, since the dataset on which the Classification Model was trained, was built by only three people, its limited size adversely affected the training of the NN. In comparison, the sentence transformer embeddings made the input to the classifier clearer as the embedding was more meaningful and its vector was larger. As for the implementation with the sentence embedding paired with the Cosine Similarity function, the results were the best, as this function was not affected by the corpus’s size like the NN.

## 6 Future Work

Although our system works well for most customer queries, the knowledge domain of the chatbot is limited due to small dataset size. Its size can be expanded by adding more intents,

patterns and responses. In addition, run-time calculations for price of a set could be an added feature to our bot. Further, run-time scraping could be enabled to answer queries not present in the dataset. Lastly, these unknown intents could be dynamically inserted into the corpus to reduce the number of scrapings required in an unseen scenario.

## References

- [1] Reddy, D. R. 1976. Speech recognition by machine: A review. In *Proceedings of the IEEE*, 64(4), 501-531.
- [2] Harshani, L. K. M. D., Weerasooriya, W. M. A. S. B., Herath, H. M. C. S., Alahakoon, P. M. K., Kumara, W. G. C. W., and Hinias, M. N. A.2021. Development of a humanoid robot mouth with text-to-speech ability.
- [3] Even-Zohar, Y., and Roth, D. 2001. A sequential model for multi-class classification. In *arXiv preprint cs/0106044*.
- [4] Yerpude, A., Phirke, A., Agrawal, A., and Deshmukh, A.2019. Sentiment Analysis on Product Features Based on Lexicon Approach Using Natural Language Processing. In *International Journal on Natural Language Computing (IJNLC)*, 8(3), 1-15.
- [5] Goldsborough, P. 2016. A tour of tensorflow. In *arXiv preprint arXiv:1610.01178*
- [6] 2021. AI Predictions In *PwC's annual AI Predictions survey*
- [7] Weizenbaum, J. 1966. ELIZA—a computer program for the study of natural language communication between man and machine. In *Communications of the ACM*, 9(1), pp.36-45.
- [8] Wallace, R.S.2009. The anatomy of ALICE. In Parsing the turing test (pp. 181-210) In *Springer, Dordrecht*.
- [9] Shum, Hy., He, Xd. and Li, D. 2018. From Eliza to XiaoIce: challenges and opportunities with social chatbots.

- In *Frontiers Inf Technol Electronic Eng*, 19, 10–26.
- [10] Walker, M.A., Rudnicky, A.I., Prasad, R., Aberdeen, J.S., Bratt, E.O., Garofolo, J.S., Hastie, H.W., Le, A.N., Pelлом, B.L., Potamianos, A. and Passonneau, R.J. 2002, September. DARPA communicator: crosssystem results for the 2001 evaluation. In *INTERSPEECH 6*
  - [11] Hoy, M.B.2018. Alexa, Siri, Cortana, and more: an introduction to voice assistants. In *Medical reference services quarterly*, 37(1), pp.81-88.
  - [12] Zhou, L., Gao, J., Li, D. and Shum, H.Y.2020. The design and implementation of xiaoice, an empathetic social chatbot. In *Computational Linguistics*, 46(1), pp.53-93.
  - [13] Dong, L., Xu, S., and Xu, B. 2018, April. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*(pp. 5884-5888). IEEE.
  - [14] Singh, J., Joesph, M. H., and Jabbar, K. B. A.2019, May. Rule-based chatbot for student enquiries. In *Journal of Physics: Conference Series*, Vol. 1228, No. 1, p. 012060.
  - [15] Ashraf, Javed and Rao, Naveed and Khattak, Naveed and Mohsin, Athar. 2010. Speaker Independent Urdu Speech Recognition Using HMM. Natural Language Processing and Information Systems. In 6177. 140-148. [10.1007/978-3-642-13881-2\\_14](https://doi.org/10.1007/978-3-642-13881-2_14).
  - [16] Bashir, Muhammad Farrukh and Javed, Abdul Rehman and Arshad, Muhammad Umair and Gadekallu, Thippa Reddy and Shahzad, Waseem and Beg, Mirza Omer2022. Context Aware Emotion Detection from Low Resource Urdu Language using Deep Neural Network. In *Transactions on Asian and Low-Resource Language Infor-*

- mation Processing.*, Vol. 4, No. 2, pp. pp.883-902.
- [17] Javed, Muhammad Saad and Majeed, Hammad and Mujtaba, Hasan and Beg, Mirza Omer2021. Fake reviews classification using deep learning ensemble of shallow convolutions. In *Journal of Computational Social Science.*, Vol. 4, No. 2, pp.883-902.
- [18] Awan, Mubashar Nazar and Beg, Mirza Omer2021. TOPrank: a topicalpostionrank for extraction and classification of keyphrases in text. In *Journal of Computational Social Science.*, Vol. 65, pp.101-116.
- [19] Qamar, Saira and Mujtaba, Hasan and Majeed, Hammad and Beg, Mirza Omer2021. Relationship Identification Between Conversational Agents Using Emotion Analysis. In *Cognitive Computation*, pp.1-15.
- [20] Javed, Abdul Rehman and Sarwar, Muhammad Usman and Beg, Mirza Omer and Asim, Muhammad and Baker, Thar and Tawfik, Hissam2020. A collaborative healthcare framework for shared healthcare plan with ambient intelligence. In *Human-centric Computing and Information Sciences*, Vol. 10, No. 1, pp.1-21.
- [21] Majeed, Adil and Mujtaba, Hasan and Beg, Mirza Omer2020. Emotion detection in Roman Urdu text using machine learning. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering Workshops*, pp.125-130.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/339722959>

# Knowledge acquisition and corpus for argumentation-based chatbots

Conference Paper · January 2019

---

CITATIONS

8

READS

123

2 authors:



Lisa Andreevna Chalaguine

University College London

13 PUBLICATIONS 94 CITATIONS

[SEE PROFILE](#)



Anthony Hunter

University College London

283 PUBLICATIONS 7,752 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Framework for Computational Persuasion [View project](#)

# Knowledge Acquisition and Corpus for Argumentation-Based Chatbots

Lisa Andreevna Chalaguine and Anthony Hunter

Department of Computer Science  
University College London, London, UK  
[{ucab1c3,a.hunter}@ucl.ac.uk](mailto:{ucab1c3,a.hunter}@ucl.ac.uk)

**Abstract.** Many of the conversations we have every day involve exchanges of arguments and counterarguments. In the context of artificial intelligence and argumentation theory, such phenomena fall into the area of dialogical argumentation. Conversational agents, also known as *chatbots*, are versatile tools that have the potential of being used in dialogical argumentation. We can assume that a chatbot would take a particular stance in the dialogue, opposing the stance of the user. In order to succeed, the chatbot also needs to be aware of various arguments and the interplay between them. Such knowledge can be represented by a directed graph, where nodes stand for arguments and arcs symbolise conflicts between them. The chatbot must be aware of both sides of the discussion, i.e. the arguments that it can play as well as ones that the user might have, to be able to formulate convincing responses. The availability of large argument graphs for research, however, is very limited. This means that researchers do not have corpora available which hinders the development of new chatbots and limits the effectiveness of existing ones. In this paper, we propose a method to acquire a large number of arguments in a graph structure using crowd sourcing. We evaluate this method in a study with participants and present a corpus which can be used for further research in computational argumentation and chatbot technologies for argumentation.

**Keywords:** argument acquisition · computational argumentation · automated chatbot knowledge acquisition · argument graphs · argument corpus

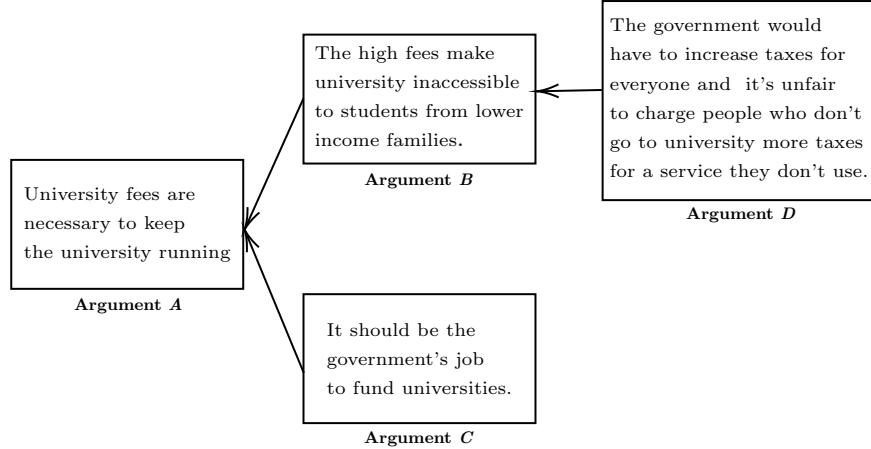
## 1 Introduction

The purpose of argumentation is to exchange different viewpoints or opinions, handle conflicting information and make informed decisions. The importance of argumentation has lead to the development of computational models of argument that aim to formalise aspects of argumentation within software. A key role for argumentation is in persuasion, and computational persuasion incorporates computational models of argument in software agents that can persuade people. This can be potentially valuable in roles such as behaviour change where the aim is to get the persuadee to make specific changes to the lifestyle (e.g. to eat

more fruit, to take more exercise, to commute by cycle, etc.) that can benefit them or those around them [1].

This calls for the development of methods for acquiring appropriate arguments and counterarguments that can be used as the chatbot's knowledge base. A situation involving argumentation can be represented by a directed graph, as proposed by Dung [2]. Each node represents an argument, and each arc denotes an attack by one argument on another. Such a graph can then be analysed to determine which arguments are acceptable according to some general criteria [3,4]. Figure 1 shows such an argument graph and the attack relationships between the arguments.

**Fig. 1.** Simple argument graph with arguments *B* and *C* attacking argument *A* and argument *D* attacking argument *C*.



Argument graphs are extensively studied in the computational argumentation literature, their acquisition, however, tends to be neglected. In order to have good quality dialogues, it is important that the argument graph has sufficient depth and breadth of coverage of the topic, so that the dialogue can proceed with more than one or two exchanges of argument per participant [5].

In order to construct graphs using *real* arguments as opposed to made-up examples, arguments have to be acquired from real-life sources. This introduces the problem of where to obtain the relevant arguments for the argument graph. This highly depends on the topic and domain in question. In the behaviour change domain, for example, arguments on why eating a lot of fruits and vegetables is healthy, may be easily found in the professional healthcare literature. Arguments on why people do not follow a healthy diet, however, have to be obtained from people directly. In politics, arguments on why a new airport is necessary, will be advertised by the government, but again, counterarguments will have to be acquired from the people who oppose that project. On other topics, arguments

of both sides may be available in either the literature or the internet. Nevertheless, these arguments have to be extracted either manually or by the means of *argument mining* and somehow organised into an argument graph.

The creation of an argument graph for a chatbot knowledge base used for dialogical argumentation raises further issues, like (1) how to capture the majority of possible arguments without making the graph too big (in order to reduce search time to make the graph usable for a chatbot which has to reply fast to avoid irritating the user), (2) which arguments to include in the knowledge base and how to justify the inclusion of some and exclusion of others (e.g. noise and repetition of arguments), and (3) how to establish relations between arguments (the arcs of the graph). In order to address these questions, a corpus is needed which can be used for experiments.

Using forums for online discussions as source for chatbot knowledge base generation (for the rest of the paper we will assume that the chatbot will be used for dialogical argumentation) sounds tempting due to the large repositories which contain a great deal of human knowledge on many topics. However, using threads from websites like *reddit* for a chatbot knowledge base raises several problems. Firstly, unless it is a very popular topic it can take months to acquire a substantial number of arguments and risk not collecting any at all. Secondly, not all posts contain arguments. Often people share stories, ask or answer questions or make opinionated statements. Thirdly, long posts most likely contain several arguments and individual arguments would therefore have to be extracted with argument mining techniques. Lastly, the resulting graph is most likely to be very imbalanced. [6] graphically shows one of the largest reddit threads which contained over 33k comments. One can see that several branches continue for quite some time before branching out further into subbranches and some of the subbranches “die” rather quickly. This kind of structure is forced by the nature of the forum exchanges and the temporal and popularity aspects of the discussion. The resulting graph may therefore be rather deep but may not have sufficient breadth, thus still requiring extension from other sources.

### 1.1 Existing Approaches

Most chatbots are implemented using templates: for a specific question the chatbot provides an answer from a list of possible answers. These are usually hand coded and the construction of chatbot knowledge bases are therefore time consuming and difficult to adapt to new domains. There is limited research on fully automated chatbot knowledge acquisition. The most relevant to our research was proposed in [7]. It describes a method of using online discussion forums to extract chatbot knowledge, by automatically extracting the titles of threads and their replies, creating <thread-title, reply> pairs. In this way a knowledge base for a chatbot is constructed. These pairs, however, are not connected in a graph like structure and the chatbot’s purpose was to answer questions and not engage in an argumentative dialogue. Chatbots that do make use of argumentation, usually assume an existing knowledge base where the counterarguments can be drawn from, or require researching the arguments and manually construct the

knowledge base. Climebot [8] (a conversational agent able to explain issues related to global warming), for example, relies on textual entailment to identify the best answer for a statement given by a human agent. The argumentative corpus from which the chatbot could choose from was extracted from three debating sites.

In our previous work [9] the arguments that the chatbot used were crowd sourced. The chatbot, however, was not aware of the users' counterarguments and was therefore not able to counter them, but only to present a new one which was not an attack to the user's argument. Hence, the chatbot was only able to acquire argument-counterargument pairs. The resulting argument graph would have extensive breadth but not go beyond two levels: the chatbot's arguments and the user's counterarguments.

A lot of research has been conducted on how to acquire arguments from the web and is generally referred to as *argument mining*. Argument mining exploits existing, and develops new, techniques from Machine Learning (ML) and Natural Language Processing (NLP); re-purposing and extending them to identify argument structures within text [10]. For an extensive overview on the latest research please refer to [11, 12]. Online generated discourse in forums or specific debating websites (e.g. *createdebate*<sup>1</sup> or *reddit*<sup>2</sup>) has also attracted research on argument mining [13, 14]. Threads from *reddit*, for example, have been used to create argument graphs for highlighting only the relevant arguments involved in a discussion [15] and assessment of persuasiveness [16]. IBM's *Debater* project [17] heavily relies on argument mining techniques and mine the arguments from published sources like Wikipedia. However, using forums for online discussions or published sources like Wikipedia as chatbot knowledge base for dialogical argumentation has its limitations, as already outlined above.

A more recent example of a chatbot that engages in dialogical argumentation is presented in [18] where the chatbot tries to persuade the user to cycle more. The chatbot's knowledge base was stored as an argument graph. The researchers undertook a web search on the pros and cons of city cycling and manually identified a number of arguments and attacks between them, which they encoded into an argument graph. Another example by the same researchers is presented in [19] on the topic of university fees in the UK which also involved a hand-crafted argument graph.

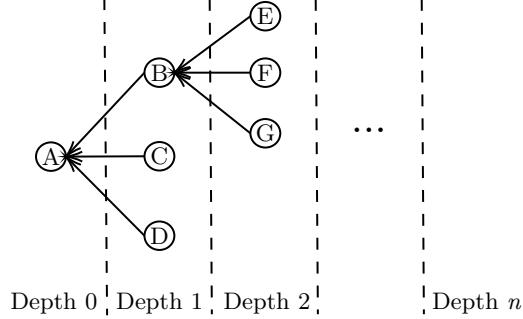
Another approach on how to collect arguments and construct an argument graph, without the use of online discussion forums or extensive research, was conducted using Dialog-Based Online Argumentation (D-BAS) and is described in [20]. Their resulting graph contains 265 arguments. It should be noted, however, that the researchers instructed the participants on how to counter previous arguments in order to obtain high-quality arguments and counterarguments. They also did not allow the repetition of arguments and motivated the participants to flag repetitions, as well as statements that should be revised, were off-topic or irrelevant, or abusive.

---

<sup>1</sup><http://www.createdebate.com/>

<sup>2</sup><http://www.reddit.com/>

**Fig. 2.** Representation of depths and attack relationships between arguments in our argument graph. Arguments  $B$ ,  $C$  and  $D$  are counterarguments to  $A$ .



## 1.2 Proposed Solution

Our aim was to generate a corpus of arguments in a graph-like structure which we could use as a chatbot knowledge base in our further research where the chatbot would engage in an argumentation dialogue with real participants. In this paper, we propose a method to acquire a large number of arguments in a graph structure using crowd sourcing and present a corpus which can be used for further research in the computational argumentation domain. Apart from a minimum and maximum length, participants had no constraints when submitting arguments in order to create a big graph of natural language arguments.

In the rest of the paper, we describe our method to create an argument corpus on the topic of university fees in the UK and evaluate the quality of the obtained arguments in an experiment with crowd sourced participants.

## 2 Method

The depth of a graph is defined as the maximum number of arcs one can follow starting from the root. We decided to create a graph of depth 5, the root argument being depth 0. Starting from the root and following any path one will end up with a maximum of 5 arguments (excluding the root argument). The arguments in depth 1 attack the root argument and are therefore *against* keeping the university fees, the arguments in depth 2 attack the arguments in depth 1 and are therefore *for* keeping the fees and so on. Figure 2 shows a schematic representation of depths in our argument graph.

In the following, we first present our method of acquiring an argument graph and then describe the acquisition of our argument graph on UK university fees using our method.

### 2.1 Argument Processing

To address the problems above we opted for using *crowd sourcing* as a means to obtain the arguments for the argument graph. For the first level (i.e. depth 1)

participants are crowd sourced and presented with the root argument in a survey and asked to counter it with a number of arguments. The resulting collection of arguments in depth 1 are all counterarguments to the root argument.

In the following, we describe a pipeline that allows to automatically extract the best arguments from the ones crowd sourced in each depth in order to include them in the graph and collect their counterarguments in the next level.

**1. Argument Length** We want a potential chatbot to give counterarguments that are neither too short, nor too long. We therefore remove all arguments that are below 15 and above 50 words in length. We would not want a potential chatbot to give a short statement as a counterarguments to the user’s argument. We do not include arguments longer than 50 words because these likely contain several arguments and we also do not consider them suitable for a chatbot knowledge base (imagine a chatbot replying with a whole paragraph).

**2. Choice of topic words** We then extract the most common words from the data (excluding stop words and words that do not add value in the given domain). The definition of *most common* depends on the size and nature of the data and is therefore up to the researcher to decide.

From the most common words, we then select *topic words* which are words which we consider meaningful in the given context. The choice of suitable topic words depends entirely on the domain and their choice is also left to the researchers’ discretion. For example, in a set of arguments on university fees, the word *money* appeared many times. It is, however, not very meaningful, whereas the words *debt* and *affordable* tell us more about the topic of the arguments. So by inspecting the frequently occurring words, the researcher can apply their knowledge of the domain to decide which would be good topic words. All arguments that contain at least one topic word are kept, the rest are removed. It should be noted that the list of topic words increases with each depth. The threshold as to how often a word has to appear in order to be considered “common” also rises since the number of arguments increases with each depth.

**3. Spell-check** We keep all arguments that contain no spelling mistakes. This can be checked by using *Grammarly*<sup>3</sup>. We delete all arguments where Grammarly highlights a typo in order to avoid including arguments into the chatbot knowledge base that contain spelling mistakes since this could influence the persuasive power of the argument. However, we do not consider incorrect punctuation or missing capitalisation as spelling mistakes, given the informality of the setting. Unfortunately, there is no Grammarly API as of the time of writing, and we therefore had to copy-paste the arguments into the Grammarly app.

**4. Final Selection of arguments for current depth** The arguments that are left after steps 1-3 are presented to crowd sourced participants who are

---

<sup>3</sup><https://app.grammarly.com/>

instructed to select those arguments that they find communicate their message the best. We opted for this wording since we were not interested in the message of the arguments (e.g. its believability or convincingness) but still want to include clear, understandable and appropriate arguments in our graph. The highest-ranked arguments are then included in depth 1 of the argument graph.

**Subsequent levels of depth** In order to minimise the need for crowd sourcing in Step 4 and in subsequent levels of depth we only keep arguments that covered (i.e. contained) the highest number of topic words. We only present arguments to crowd sourced participants for ranking, where the topic words are the same and a selection has to be made. This ranking is as in step 4 where we ask the participants which arguments communicate their message the best. This way the need for participants in Step 4 is reduced significantly after depth 1. The idea behind this method is to include arguments in the argument graph that address the maximum number of issues as represented by the topic words.

## 2.2 Argument Acquisition for Next Depth

The arguments for all subsequent levels were collected by presenting the arguments from the previous level to crowd sourced participants who were asked to counter them. Steps 1-3 are then applied to the collected arguments for that level. The participants were presented the last two arguments in the graph since presenting only the last may be confusing without the attacked one as a reference. For example, during the acquisition of arguments in depth 4, participants are shown the argument from depth 2, one of its counterarguments in depth 3 and asked to assume the stance of the argument in depth 2 and counter the argument in depth 3.

## 3 Case Study and Corpus

In the UK, the current situation is that home students (students from the EU, including the UK) pay around 9000£ tuition fees per year for a Bachelor's degree. This is a controversial situation, with supporters and contestants on both sides. We therefore chose this as a suitable topic for our task and selected "*Universities in the UK should continue charging students the 9k tuition fee per year*" as the root topic for our graph. In the following, we describe how we acquired our argument graph corpus on university fees in the UK applying our method described above.

Participants were recruited via *Prolific*<sup>4</sup>, which is an online recruiting platform for scientific research studies, and were paid for taking part in our study. We used Google Forms for our study. The prerequisites for taking part in the study were to be over 18, fluent in the English language and a current resident

---

<sup>4</sup><https://www.prolific.ac/>

of the UK (in order to minimise the risk of recruiting participants who do not know anything about university fee situation in the UK).

For depth 1 we recruited 91 participants who were asked to provide 3 different reasons in a Google Form on why they think that the 9k tuition fees in the UK were not appropriate and should be abolished. We therefore collected 273 (3 x 91) arguments at depth 1.

Many responses consisted of short statements like “*It is too expensive*” or “*students are poor people*” which we would not want a potential chatbot to give as counterarguments to the user’s argument. During the argument acquisition in future depths we instructed the participants to provide arguments that were at least 15 words in length as we were only left with 97 arguments after this step in depth 1<sup>5</sup>.

We then extracted the most common words from our data. We mentioned above that we delete stopwords and words that do not add value in the given domain from our data. In our case these were words like *education*, *university*, *fee*, *abolish*, *students*, *degree* and *tuition*. We extracted all words that came up at least 5 times in the dataset of 97 arguments.

From the most common words we selected the words *job*, *debt*, *afford/ affordable*, *access/accessible* and *free* as topic words for depth 1. Other common words included *study*, *high*, *amount*, *money*, *pay* and *work*, which we believed were too general. We mentioned above that the list of topic words grows with each depth: In depth 2, for example, the words *loan*, *tax*, *government* and *scholarship* were added to the list of topic words.

After steps 1-3 we were left with 48 arguments out of the 273 at depth 1. At depth 1 we decided to include 3 arguments for each topic word in the graph. We created 5 surveys (one for each topic word) which presented all arguments that included the topic word in question. We crowd sourced 20 participants per survey and instructed them that the arguments might be very similar and all touch on a certain aspect but that the individual arguments differ in their quality. We asked them to select those arguments that they found communicate their message the best. We then used the three arguments that were ranked the highest in each group (some arguments contained two topic words, therefore some topic words are represented by more than 3 arguments).

Our aim was to create a graph where each argument after depth 1 has 3 counterarguments (on average) to avoid making the graph too big and due to limited funding. In subsequent depths we only kept arguments that covered the highest number of topic words. Only if the topic words of several arguments were the same and a selection had to be made those arguments were presented to crowd sourced participants for ranking.

For example, consider an argument in depth 1 that had 6 counterarguments in depth 2 after applying Steps 1-3. The counterarguments (CA) contained the following topic words:

---

<sup>5</sup>When the study took place Google Forms did not support response validation. Since July 2019 a minimum character count can be specified.

<b>CA 1</b> loan, debt	<b>CA 4</b> government
<b>CA 2</b> loan, debt, scholarship	<b>CA 5</b> loan
<b>CA 3</b> loan, government	<b>CA 6</b> loan, government

CA 1 and CA 2 were selected for the next depth because they contained the highest number of topic words and no other CAs contained the same combination. CA 3 and CA 6 were presented in a survey to participants in order to choose the “better” one for the graph. CA 4 and CA 5 were not selected because all other CAs contained at least one of the topic words of CA 4 and CA 5.

Depth 1 consists of 16 arguments. We created 3 surveys (containing of 5, 5 and 6 arguments respectively) and recruited 10 participants per surveys to counter the given arguments. We split the arguments into three smaller surveys in order to avoid presenting similar arguments and reduce the risk of participants giving the same counterargument to several arguments. For each subsequent level of depth, the arguments from the previous depth were divided into surveys of 5-6 arguments and 10 participants were recruited per survey. We therefore acquired 10 counterarguments per argument in each depth. Participants were presented the last two arguments in the graph. For example, during the acquisition of arguments in depth 4, participants were shown the argument from depth 2 (against fees), one of its counterarguments in depth 3 (pro fees) and asked to assume the position of being against fees and counter the argument in depth 3. It should be noted that for depth 5 we only recruited 5 participants to counter the arguments of depth 4.

### 3.1 The Corpus

Our graph contains 1288 arguments with each argument on average having 3 counterarguments, and consists of 5 depths making it the most extensive corpus of this kind. The overall corpus of acquired arguments contains over 4000 arguments.

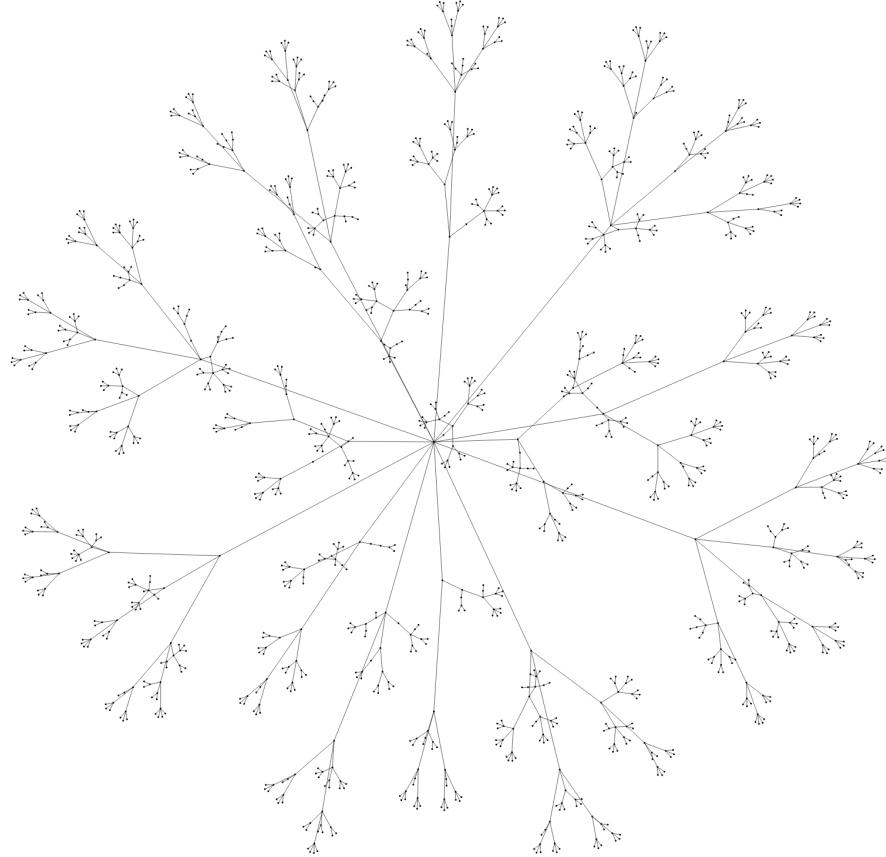
The generated corpus can be found on github [21]. It consists of two data sets. One data set contains the raw arguments acquired for each depth. The second dataset contains the arguments that were used in the generation of the argument graph. Each argument contains a unique ID and the ID of the attacked argument in the previous depth. For example, an argument in depth 2 may have the id *depth2\_6* and the ID of the attacked argument *depth1\_34* which means argument *depth2\_6* attacks argument *depth1\_34*.

The github repository also contains the `python` code to generate a visual network graph using the `pyvis` library. The resulting visualisation displays the arguments when hovering over the nodes and is shown in Figure 3 (a higher resolution picture is available in the github repository) [21].

## 4 Evaluation

We evaluated our generated argument corpus by randomly creating 24 dialogues by following the arcs of the graph, starting from the root and following each of

**Fig. 3.** Visualisation of the generated argument corpus in graph form



the 16 arcs from the root to the argument in depth 1 at least once. This way we ensured to create at least 16 completely distinct dialogues. We divided the 24 dialogues into 4 surveys using Google Forms and recruited 20 participants for each survey to judge the 6 given dialogues. An example dialogue is given below.

PERSON A: *Universities in the UK should continue charging students the 9k tuition fee per year.*

PERSON B: *Education should be available for everyone, not for only ones who can afford it.*

PERSON A: *People who can't afford have government help. Government can't afford free education for all unless they increase the taxes and people won't like it.*

PERSON B: *The government are still paying for the loans and probably won't see the money back when the loans are written off in 30 years time. Cheaper education and higher taxes is more sustainable than relying on students to pay back the loans, which they won't.*

PERSON A: *The government should step out then and leave it to the banks to take the risk. Anyway with higher taxes and cheap education there would be plenty of educated unemployed to pay by the Government.*

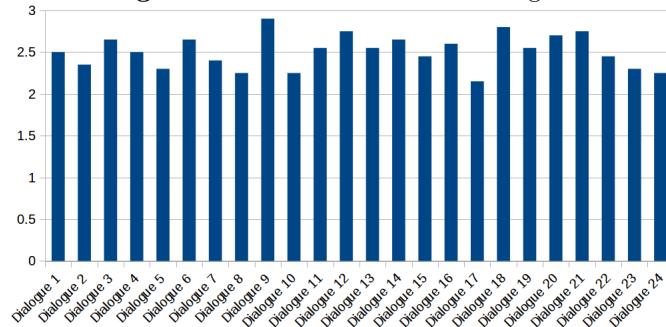
PERSON B: *Banks would likely impose even higher rates of interest which would be unsustainable and they may also reject a large number of students given their financial circumstances.*

We informed the participants that the study involved judging transcripts and that the given dialogues involved two parties arguing whether tuition fees in the UK should be kept at 9000£. Party A believes they should be kept and Party B believes they should be abolished. We instructed them to judge 6 transcripts plus an additional one playing the role of an attention check to ensure honesty/quality of the provided responses. We asked them to score the transcripts in respect of each party staying to the point and defending their point of view. We asked them to not judge the dialogues by whether they believed the presented arguments since we were only interested in the overall quality of the dialogue (whether they make sense and parties sticking to their point of view). The participants were given a choice of three:

1. *Both parties don't stick to the point and don't defend their point of view*
2. *Both parties somewhat stick to the point and somewhat defend their point of view*
3. *Both parties do stick to the point and do defend their point of view*

On average each dialogue scored 61% for option 3 (both parties sticking to the point and defending their point of view), 29% for option 2, and only 10% for option 1. Figure 4 shows the score for each dialogue, option 1 (*don't*) receiving score 1, option 2 (*somewhat*) receiving score 2 and option 3 (*do*) receiving score

**Fig. 4.** Scores for each individual dialogue.



3. The average score per dialogue was 2.51 which shows that the dialogues were of good quality and that if following a path in the graph, the resulting dialogue makes sense despite the individual arguments being collected from different people.

## 5 Discussion and Conclusions

In this paper, we introduce a methodology to acquire a corpus of arguments for dialogues and present a corpus for research for computational argumentation, natural language processing, and chatbot knowledge base construction. Apart from checking for spelling mistakes, we have not conducted any further quality assessment of the arguments and have not checked for duplicate arguments in the argument graph. This gives researchers the possibility to use our corpus for research in methods like:

- Argument similarity assessment [22, 23]: many arguments in the graph support the same idea and are fairly similar. However, one can say the same thing in completely different ways, and clustering arguments by their similarity is a challenging but potentially valuable task.
- Argument quality assessment [24–26]: After clustering similar arguments together one could apply some sort of quality assessment in order to decide which argument in the cluster is the “best” according to some criteria (e.g. convincingness [27]).
- Establishing more attack (and support) relationships between arguments in the graph [28, 29]: After identifying similar arguments one could establish more attack relationships in the graph. For example, if arguments A and B are the same (just differently phrased), the counterarguments of A also attack B and vice versa.

By applying the methods above high-grade chatbot knowledge bases could be created that contain only arguments of the highest quality (however one chooses to assess that) and contain a high number of possible arguments for that domain.

We also evaluated the quality of our corpus and believe that publishing it will give researchers a resource to explore the topics mentioned above, which will facilitate further research in these areas.

In future work we want to create a chatbot that uses our generated argument graph as knowledge base and use it in a study with real participants. The participants could be on either side of the debate (either for or against keeping university fees) and the chatbot would defend the opposite standpoint. In order to evaluate our chatbot, the participants could be asked to judge the chat with the chatbot on persuasiveness and other metrics like the quality of the dialogue and whether the chatbot gave relevant replies (counterarguments).

## 6 Acknowledgments

The first author is funded by a PhD studentship from the EPSRC. The authors would like to thank Sylwia Polberg for valuable feedback on earlier versions of this paper.

## References

1. A. Hunter. Computational persuasion with applications in behaviour change. In *Proc. of Computational Models of Argument 2016*, pages 5–18, 2016.
2. P.M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.
3. P. Besnard, A. Javier García, A. Hunter, S. Modgil, H. Prakken, G. Simari, and F. Toni. Introduction to structured argumentation. *Argument and Computation*, 5(1):1–4, 2014.
4. P. Baroni, M. Caminada, and M. Giacomin. An introduction to argumentation semantics. In *Knowledge Engineering Review* 26(4), pages 365–410, 2011.
5. A. Hunter, L. Chalaguine, T. Czernuszenko, E. Hadoux, and S. Polberg. Towards computational persuasion via natural language argumentation dialogues. In *Proc. of Kuenstliche Intelligenz 2019 (in press)*, 2019.
6. Reddit thread. <https://tinyurl.com/y267p2lq>.
7. J. Huang, M. Zhou, and D. Yang. Extracting chatbot knowledge from online discussion forums. In *Proc. of the 20th International Joint Conference on Artificial Intelligence*, pages 423–428, 2007.
8. D. Toniuc and A. Groza. Climebot: An argumentative agent for climate change. In *Proc. of the 2017 IEEE 13th International Conference on Intelligent Computer Communication and Processing*, pages 63–70, 2017.
9. L. A. Chalaguine, A. Hunter, F. L. Hamilton, and H. W. W. Potts. Impact of argument type and concerns in argumentation with a chatbot. In *Proc. of the 31st International Conference on Tools with Artificial Intelligence 2019 (in press)*, 2019.
10. S. Wells. Argument mining: Was ist das? In *Proc. of the 14th International Workshop on Computational Models of Natural Argument*, 2014.
11. E. Cabrio and S. Villata. Five years of argument mining: a data-driven analysis. In *Proc. of the 27th International Joint Conference on Artificial Intelligence*, pages 5427–5433, 2018.
12. M. Lippi and P. Torroni. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology* 16(2), pages 1–25, 2016.
13. I. Habernal and I. Gurevych. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179, 2017.
14. R. Swanson, B. Ecker, and M. Walker. Argument mining: Extracting arguments from online dialogue. In *Proc. of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 217–226, 2015.
15. A. Pazienza, S. Ferilli, and F. Esposito. Constructing and evaluating bipolar weighted argumentation frameworks for online debating systems. In *Proc. of the 1st Workshop on Advances In Argumentation In Artificial Intelligence*, pages 111–125, 2017.

16. C. Tan, V. Niculae, C. Danescu-Niculescu-Mizil, and L. Lee. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proc. of the 25th International Conference on World Wide Web*, pages 613–624, 2016.
17. R. Levy, B. Bogin, S. Gretz, R. Aharonov, and N. Slonim. Towards an argumentative content search engine using weak supervision. In *Proc. of the 27th International Conference on Computational Linguistics*, pages 2066–2081, 2018.
18. E. Hadoux and A. Hunter. Comfort or safety? Gathering and using the concerns of a participant for better persuasion. 2019.
19. A. Hunter, S. Polberg, and E. Hadoux. Strategic argumentation dialogues for persuasion: Framework and experiments based on modelling the beliefs and concerns of the persuadee. Technical report.
20. T. Krauthoff, C. Meter, and M. Mauve. Dialog-based online argumentation: Findings from a field experiment. In *Proc. of the 1st Workshop on Advances in Argumentation in Artificial Intelligence*, pages 85–99, 2017.
21. Corpus: [https://github.com/lisanka93/Argument\\_Graph\\_Corpus](https://github.com/lisanka93/Argument_Graph_Corpus).
22. F. Boltuzic and J. Snajder. Identifying prominent arguments in online debates using semantic textual similarity. In *Proc. of the 2nd Workshop on Argumentation Mining*, pages 110–115, 2015.
23. A. Misra, B. Ecker, and M. A. Walker. Measuring the similarity of sentential arguments in dialog. In *Proc. of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 276–287, 2016.
24. H. Wachsmuth, S. Syed, and B. Stein. Retrieval of the best counterargument without prior topic knowledge. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, 2018.
25. H. Wachsmuth, N. Naderi, I. Habernal, Y. Hou, G. Hirst, I. Gurevych, and B. Stein. Argumentation quality assessment: Theory vs. Practice. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 250–255, 2017.
26. H. Wachsmuth, N. Naderi, Y. Ho, Y. Bilu, V. Prabhakaran, A. T. Thijm, G. Hirst, and B. Stein. Computational argumentation quality assessment in natural language. In *Proc. of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 176–187, 2017.
27. I. Habernal and I. Gurevych. Which argument is more convincing? Analyzing and predicting convincingness of web arguments using bidirectional LSTM. In *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1589–1599, 2016.
28. O. Cocarascu and F. Toni. Identifying attack and support argumentative relations using deep learning. In *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1374–1379, 2017.
29. L. A. Chalaguine, A. Hunter, F. L. Hamilton, and H. W. W. Potts. Argument harvesting using chatbots. In *Proc. of Computational Models of Argument 2018*, pages 149–160, 2018.

# “Mama Always Had a Way of Explaining Things So I Could Understand”: A Dialogue Corpus for Learning to Construct Explanations

**Henning Wachsmuth \***

Department of Computer Science  
Paderborn University  
henningw@upb.de

**Milad Alshomary \***

Department of Computer Science  
Paderborn University  
milad.alshomary@upb.de

## Abstract

As AI is more and more pervasive in everyday life, humans have an increasing demand to understand its behavior and decisions. Most research on explainable AI builds on the premise that there is one ideal explanation to be found. In fact, however, everyday explanations are co-constructed in a dialogue between the person explaining (the explainer) and the specific person being explained to (the explainee). In this paper, we introduce a first corpus of dialogical explanations to enable NLP research on how humans explain as well as on how AI can learn to imitate this process. The corpus consists of 65 transcribed English dialogues from the Wired video series *5 Levels*, explaining 13 topics to five explainees of different proficiency. All 1550 dialogue turns have been manually labeled by five independent professionals for the topic discussed as well as for the dialogue act and the explanation move performed. We analyze linguistic patterns of explainers and explainees, and we explore differences across proficiency levels. BERT-based baseline results indicate that sequence information helps predicting topics, acts, and moves effectively.

## 1 Introduction

Explaining is one of the most pervasive communicative processes in everyday life, aiming for mutual understanding of the two sides involved. Parents explain to children, doctors to patients, teachers to students, seniors to juniors—or all the other way round. In explaining dialogues, one side takes the role of the *explainer*, the other the role of the *explainee*. Explainers seek to enable explainees to comprehend a given topic to a certain extent or to perform some action related to it (Rohlfing et al., 2021). This usually implies a series of dialogue turns where both sides request and provide different information about the topic. In line with the quote from the movie “Forrest Gump” in the title,

Explaining dialogue on the main topic “blockchain”

- |    |   |
|----|---|
| 01 | Do you know what we're gonna talk about today? It's called blockchain.  |
| 02 | What's blockchain?  |
| 03 | That's a really good question. It's actually a way that we can trade. Do you know what trade is?  |
| 04 | Mmm-hmm, it's when you take turns doing something. It's when you give up most of what you want, right?  |
| 05 | When you give up most of what you want? Well, sometimes that definitely happens for sure. What if I told you that this is the kind of technology that I work on that means you could trade with any kid all over the world?   |
| 06 | Really?   |
| 07 | Yeah.   |
| 08 | If I could trade with any kid, I would trade, well, I would trade something I don't like so much.   |
| 09 | That's probably a good idea, maybe somebody else likes it more than you do. So normally, when people trade, they have to go to the store, or they have to know the person so they can get what they asked for. With blockchain, you can make that exact same trade, but you don't need the store, and you don't even necessarily need to know the other person. |
| 10 | Really?   |
| 11 | Really.   |

Explainer (expert)

(child) Explainee

Figure 1: A short explaining dialogue from the video series *5 Levels*, included in the corpus presented in Section 3. Here, an expert explains blockchain to a child.

how an explaining dialogue looks like is strongly affected by the specific explainer and explainee as well as by their interaction.

Consider the dialogue in Figure 1, where a technology expert explains the basic idea of blockchain to a 5-year old in a controlled setting. Beyond the explanations of the main topic (turns 05 and 09), the dialogue contains an explanation request (02), a test of prior knowledge (03), explanations from the explainee (04), and more. We observe that the explainer’s explanations depend on the reaction of the explainee and that their level of depth is most likely adjusted to the explainee’s proficiency.

The importance of studying how to explain has become apparent with the rise of research on explainable artificial intelligence, XAI (Barredo Arrieta et al., 2020). As AI finds its way into various

\* Both authors contributed equally to this paper.

aspects of work and private life, humans interacting with respective systems, or being affected by them, have an increasing demand to understand their behavior and decisions. This demand has also been manifested in a *right to explanation* within the EU’s General Data Protection Regulation (Goodman and Flaxman, 2017). Prior work on XAI largely starts from the premise that an ideal (monological) explanation exists for any behavior or decision, possibly dependent on the explainee at hand (Miller, 2019). According to Rohlfing et al. (2021), however, real explainability must account for the co-constructive nature of explaining emerging from interaction.

In natural language processing, early work modeled discourse structure of monological explanations (Bourse and Saint-Dizier, 2012), and a number of recent approaches generate respective explanations for XAI (Situ et al., 2021) and recommendation (Li et al., 2021). In contrast, the language of dialogical explanations is still understudied (details in Section 2). We argue that a better understanding of how humans explain in dialogues is needed, so that XAI can learn to interact with humans.

In this paper, we present a first corpus for computational research on how to explain in dialogues (Section 3). The corpus has been created as part of a big interdisciplinary research project dealing with the construction of explainability.<sup>1</sup> It consists of 65 transcribed dialogical explanations from the American video series *5 Levels* freely published by the Wired magazine.<sup>2</sup> Five dialogues each refer to one of 13 science-related topics (e.g., “blockchain” or “machine learning”). They have the same explainer (an expert on the topic), but differ in the explainee’s proficiency (from child to colleague).

To enable XAI to mimic human explainers, it has to learn what turn to make at any point in a dialogue. In discussion with humanities researchers, we model a turn for this purpose by the relation of its *topic* to the main topic (e.g., subtopic or related topic), its *dialogue act* (e.g., check question or informing statement), and its *explanation move* (e.g., testing prior knowledge or providing an explanation). We segmented the dialogues into a total of 1550 turns, and we let five independent professionals annotate each turn for these three dimensions.

In Section 4, we analyze linguistic patterns of explaining dialogues in the annotated corpus. We find clear signals for the explainer’s alignment to

the explainee’s proficiency, such as the avoidance of deviating to related topics towards children. The roles of explainer and explainee are reflected in the varying use of dialogue acts and explanation moves, possibly stressed by the given setting.

To obtain baselines for the prediction of the three annotated dimensions, we evaluate three variants of BERT (Devlin et al., 2019) in 13-topic cross-validation on the corpus (Section 5). Our results reveal that modeling sequential dialogue interaction helps predicting a turn’s topic, act, and move effectively. Improvements seem still possible, calling for more sophisticated approaches as well as for more explaining dialogue data in the future.<sup>3</sup>

In summary, the contributions of our paper are:

1. A manually annotated corpus for studying how humans explain in dialogical settings
2. Empirical insights into how experts explain to explainees of different proficiency levels
3. Baselines for predicting the topic, dialogue act, and explanation move of dialogue turns

## 2 Related Work

Explainable AI (XAI) largely focuses on the interpretability of learned models from the perspective of scientific completeness (Gilpin et al., 2018). Even though recent works tackle cognitive aspects, such as the trade-off between completeness and compactness (Confalonieri et al., 2019), Miller (2019) pointed out that this perspective is far away from the understanding of everyday explanations in the social sciences. Garfinkel (2009) argues that the key is to sort out what the explainer should actually explain, and Barredo Arrieta et al. (2020) stressed the importance of who is the explainee for XAI. Rohlfing et al. (2021) built on these works, but reasoned that explanations can only be successful in general, if they are co-constructed in interaction between explainer and explainee. The rationale is that explainees vary in their motives and needs, and they face different challenges (Finke et al., 2022). The corpus we present serves as a basis for studying the linguistic aspects of the explainer-explainee interaction computationally.

Natural language processing (NLP) has notably dealt with the related genre of instructional texts, modeling their structure (Fontan and Saint-Dizier, 2008), extracting knowledge (Zhang et al.,

---

<sup>1</sup>Constructing Explainability, <https://trr318.upb.de/en>

<sup>2</sup>5 Levels, <https://www.wired.com/video/series/5-levels>

<sup>3</sup>The corpus and the experiment code are freely available here: <https://github.com/webis-de/COLING-22>

2012), comprehending some meaning (Yagcioglu et al., 2018), or generating them (Fried et al., 2018). However, instructional text has a clear procedural style with distinctive surface features (Vander Linden, 1992), unlike explanations in general. For tutorial applications, Jordan et al. (2006) extracted concepts from explanation sentences, whereas Jansen et al. (2016) studied the knowledge needed for scientific explanations, and Son et al. (2018) identified causal explanations in social media. Towards a computational understanding of explaining, Bourse and Saint-Dizier (2012) modeled explanation structure with discourse relations (Mann and Thompson, 1988). In XAI and recommendation contexts, the generation of respective explanations is explored increasingly (Situ et al., 2021; Li et al., 2021).

However, our main goal is not to understand how to generate an explanation, but to model how people interact in an explanation process. For annotation, we thus rely on the widely accepted concept of dialogue acts (Stolcke et al., 2000; Bunt et al., 2010). Similar has been done for deliberative dialogues by Al Khatib et al. (2018). In addition, we model the *moves* that explainers and explainees make in their interaction, adapting the idea of rhetorical moves, in terms of communicative functions of text segments used to support the communicative objective of a full text (Swales, 1990). Wachsmuth and Stein (2017) proposed task-specific moves for monological arguments, but we are not aware of any work on moves for explanations, nor for dialogical settings.

Hence, we start by compiling data in this paper. Existing related corpora contain tutorial feedback for explanation questions (Dzikovska et al., 2012), answers to non-factoid questions (Dulceanu et al., 2018), and pairs of questions and responses from community question answering platforms (Nakov et al., 2017). Finally, the corpus of Fan et al. (2019) includes 270k threads from the Reddit forum *Explain like I'm Five* where participants explain a concept asked for in simple ways. While all these allow for in-depth analyses of linguistic aspects of explanations, none of them include explaining dialogues with multiple turns on each side. This is the gap we fill with the corpus that we introduce.

### 3 Data

This section introduces the corpus that we created to enable computational research on dialogical explanation processes of humans. We discuss our

design choices with respect to the source and annotation, and we present detailed corpus statistics.

#### 3.1 Explaining Dialogues on Five Levels

As source data, we decided to rely on explaining dialogues from a controlled setting in which two people explicitly meet to talk about a topic to be explained. While we thereby may miss some interaction behavior found in real-word explanation processes, we expect that such a setting best exhibits explaining dialogue features in their pure form.

In particular, we acquired the source dialogues in our corpus from *5 Levels*, an American online video series published by the Wired magazine. In each video of the series, one explainer explains a science-related or technology-related topic to five different explainees. The explainer is always an expert on the topic, whereas the explainees increase in terms of (assumed) proficiency on the topic:

1. a *child*,
2. a *teenager*,
3. an *undergrad* college student,
4. a *grad* student, and
5. a *colleague* in terms of another expert.

Every video starts with a few introductory words by the expert, before one dialogue follows the other.<sup>4</sup> Transcriptions are already provided in the videos' captions. So far, the first season of the series is available with a total of 17 videos. Table 1 lists all explained topics (*main topics* henceforth) in these videos, along with explainer information.

At the time of starting the annotation process discussed below, only 14 of the 17 videos had been accessible, and one of these had partly corrupted subtitles. We thus restricted the annotated corpus to the remaining 13 videos, summing up to 65 dialogues that correspond to a video length of 5.35 hours. Later, we added all dialogues from the other four videos in unannotated form to the corpus.

Before annotation, we manually segmented each dialogue into its single turns, such that consecutive turns in a dialogue alternate between explainer and explainee. Overall, the 65 dialogues consist of 1550 turns (23.8 turns per dialogue on average), 790 from explainers and 760 from explainees. The turns span 51,344 words (33.1 words per turn). On

<sup>4</sup>It is noteworthy that the videos seem to have been cut a little, likely for the sake of a concise presentation. We assume that this mainly removed breaks between dialogue turns only. While it limits studying non-verbal interaction in explaining, the effect for textual analyses of the dialogues should be low.

#	Topic	Explainer	Expertise
1	Harmony	Jacob Collier	Musician
2	Blockchain	Bettina Warburg	Political scientist
3	Virtual reality	John Carmack	Oculus CTO
4	Connectome	Bobby Kasthuri	Neuroscientist
5	Black holes	Varoujan Gorjian	NASA astronomer
6	Lasers	Donna Strickland	Professor
7	Sleep	Aric A. Prather	Sleep scientist
8	Dimensions	Sean Carroll	Theoret. physicist
9	Gravity	Janna Levin	Astrophysicist
10	Computer hacking	Samy Kamkar	Security researcher
11	Nanotechnology	George Tulevski	Nanotec. researcher
12	Origami	Robert J. Lang	Physicist
13	Machine learning	Hilary Mason	Hidden Door CEO
14	CRISPR	Neville Sanjana	Biologist
15	Memory	Daphna Shohamy	Neuroscientist
16	Zero-knowl. proof	Amit Sahai	Computer scientist
17	Black holes	Janna Levin	Astrophysicist

Table 1: All 17 main topics explained in the *5 Levels* dialogues, along with the explainers and their expertise. The 65 dialogues of the 13 topics listed in black are annotated in our corpus; the rest is provided unannotated.

average, an explainer’s turn is double as long as an explainee’s turn (43.7 vs. 22.1 words). While the general data size is not huge, we provide evidence in Sections 4 and 5 that it suffices to find common patterns of explanation processes. Limitations emerging from the size are discussed in Section 6.<sup>5</sup>

### 3.2 Annotations of Explanatory Interactions

The corpus is meant to provide a starting point for XAI systems that mimic the explainer’s role within dialogical explanation processes. Our annotation scheme supports this purpose and is the result of extensive discussions in our interdisciplinary project with a big team of computer scientists, linguists, psychologists, and cognitive scientists. Where possible, we followed the literature, but the lack of research on human interaction in explaining (see Section 2) made us extend the state of the art in different respects.

In particular, we focus on turn-level category labels that capture the basic behavior of explainers and explainees in explaining dialogues. Our scheme models the three dimensions of dialogue turns that we agreed on to be needed for a computational understanding of the behavior:

- the relation of a turn’s *topic* to the main topic,
- the *dialogue act* performed in the turn, and
- the *explanation move* made through the turn.

<sup>5</sup>We also extracted the time code (start and end milliseconds) of each segment from the videos, for which one caption is shown. This may serve multimodal studies in the future.

We discuss the labels considered for each of the three annotation dimensions in the following. Since all labels apply to both explainer and explainee in principle, we refer to a speaker and a listener below.

**Topic** Even though the dialogues we target have one defined main topic to be explained, what is explained in specific turns may vary due to the dynamics of explaining interaction (Garfinkel, 2009). Since we seek to learn how to explain in general rather than any specificities of the concrete 13 main topics in the corpus, we abstract from the latter, modeling only the relation of the topic discussed in a turn to the dialogue’s main topic. In particular, a turn’s topic may be annotated as follows:

- t<sub>1</sub> *Main topic*. The main topic to be explained;
- t<sub>2</sub> *Subtopic*. A specific aspect of the main topic;
- t<sub>3</sub> *Related topic*. Another topic that is related to the main topic;
- t<sub>4</sub> *No/Other topic*. No topic, or another topic that is unrelated to the main topic.

**Dialogue Act** To model the communicative functions of turns in dialogues, we follow the literature (Bunt et al., 2010), starting from the latest version of the ISO standard taxonomy of dialogue acts.<sup>6</sup> In explaining, specific dialogue acts are in the focus, though. In collaboration with the interdisciplinary team, we selected a subset of 10 acts that capture communication on a level of detail that is specific enough to distinguish key differences, but abstract enough to allow finding recurring patterns:

- d<sub>1</sub> *Check question*. Asking a check question;
- d<sub>2</sub> *What/How question*. Asking a what question or a how question of any kind;
- d<sub>3</sub> *Other question*. Asking any other question;
- d<sub>4</sub> *Confirming answer*. Answering a question with confirmation;
- d<sub>5</sub> *Disconfirming answer*. Answering a question with disconfirmation;
- d<sub>6</sub> *Other answer*. Giving any other answer;
- d<sub>7</sub> *Agreeing statement*. Conveying agreement on the last utterance of the listener;
- d<sub>8</sub> *Disagreeing statement*. Conveying disagreement accordingly;
- d<sub>9</sub> *Informing statement*. Providing information with respect to the topic stated in the turn;
- d<sub>10</sub> *Other*. Performing any other dialogue act.

<sup>6</sup>DIT++ Taxonomy of Dialogue Acts, <https://dit.uvt.nl>

**Explanation Move** Finally, we aim to understand the explanation-specific moves that explainers and explainees make to work together towards a successful explanation process. Due to the lack of models of explaining dialogues (see Section 2, we started from recent theory of explaining (Rohlfing et al., 2021). Based on a first inspection of a corpus sample, we established a set of 10 explanation moves that a speaker may make in the process, at a granularity similar to the dialogue acts:<sup>7</sup>

- e<sub>1</sub> *Test understanding.* Checking whether the listener understood what was being explained;
- e<sub>2</sub> *Test prior knowledge.* Checking the listener’s prior knowledge of the turn’s topic;
- e<sub>3</sub> *Provide explanation.* Explaining any concept or a topic to the listener;
- e<sub>4</sub> *Request explanation.* Requesting any explanation from the listener;
- e<sub>5</sub> *Signal understanding.* Informing the listener that their last utterance was understood;
- e<sub>6</sub> *Signal non-understanding.* Informing the listener that the utterance was not understood;
- e<sub>7</sub> *Providing feedback.* Responding qualitatively to an utterance by correcting errors or similar;
- e<sub>8</sub> *Providing assessment.* Assessing the listener by rephrasing their utterance or giving a hint;
- e<sub>9</sub> *Providing extra info.* Giving additional information to foster a complete understanding;
- e<sub>10</sub> *Other.* Making any other explanation move.

We note the hierarchical nature of the scheme with respect to dialogue acts and explanations; for example, d<sub>1</sub>–d<sub>3</sub> could be merged as well as e<sub>1</sub>–e<sub>2</sub>. While some acts and moves are much more likely to be made by an explainer or an explaine, we did not restrict this to avoid biasing the annotators.<sup>8</sup>

### 3.3 Crowd-based Annotation Process

The restriction of the annotations to a manageable number of turn-level labels was also made to make the annotation process simple enough to carry it out with independent people. In particular, we hired five freelancers, working as content editors and

<sup>7</sup>We decided to leave a distinction of different explaining types (such as causal or analogy-based explanations) to future work, as it does not match the level of detail in our scheme.

<sup>8</sup>For dialogue acts d<sub>3</sub>, d<sub>6</sub>, and d<sub>10</sub> as well as explanation move e<sub>10</sub>, the annotators had to name the label in free text. We provide these as part of the corpus, we give individual examples of other moves and acts in Section 4.

annotators on the professional crowdsourcing platform *Upwork*. All were native speakers of English with a 90%+ job success rate on the platform. We clarified the task individually with each of them.

We provided guidelines based on the definitions above, along with general explanations and some examples. Using Label Studio,<sup>9</sup> we developed a task-specific user interface where each dialogue was shown as a sequence of turns and one label of each dimension could be assigned to a turn (if multiple labels seemed appropriate, the best fitting one). Each annotator labeled all 1550 turns. We paid \$ 1115 for an overall load of 85 hours, that is, \$ 13.12 per hour on average (with minor differences for annotators due to bonuses and varying durations).

**Agreement** In terms of the conservative measure Fleiss’  $\kappa$ , the inter-annotator agreement among all five was 0.35 for the topic, 0.49 for dialogue acts, and 0.43 for explanation moves. While these values indicate moderate agreement only, they are in line with related subjective labeling tasks of short texts such as news sentences (Al Khatib et al., 2016) and social media arguments (Habernal et al., 2018). Moreover, we exploited the multiple labels we have per turn to consolidate reliable annotations, as described in the following.

**Output Annotations** For consolidation, we rely on MACE (Hovy et al., 2013), a widely used technique for grading the reliability of crowdworkers based on their agreement with others. The MACE competence scores of the annotators suggest that all did a reasonable job in general, lying in the ranges 0.30–0.76 (topic), 0.58–0.82 (dialogue acts), and 0.45–0.85 (explanation moves) respectively. We applied MACE’ functionality to derive one aggregate output label for each dimension from the five annotations weighted by competence scores.

### 3.4 The Wired Explaining Dialogue Corpus

Table 2 presents detailed general statistics of the three annotation dimensions. More insights into the distribution of annotations across proficiency levels follow in Section 4.

With respect to topic (t<sub>1</sub>–t<sub>4</sub>), about half of all turns explicitly discuss the *main topic* (27.7%), a *subtopic* (5.7%), or a *related topic* (16.8%). Explainees much more often mention none of these (62.8% vs. 37.3%), underlining the leading role of the explainer in dialogue setting.

<sup>9</sup>Label Studio, <https://labelstud.io>

Label	Explainer		Explainee		Total	
	#	%	#	%	#	%
t <sub>1</sub> Main topic	<b>301</b>	<b>38.1</b>	129	17.0	430	27.7
t <sub>2</sub> Subtopic	52	6.6	36	4.7	88	5.7
t <sub>3</sub> Related topic	142	18.0	118	15.5	260	16.8
t <sub>4</sub> Other/No topic	295	37.3	<b>477</b>	<b>62.8</b>	<b>772</b>	<b>49.8</b>
d <sub>1</sub> Check question	183	23.2	62	8.2	245	15.8
d <sub>2</sub> What/How question	77	9.7	38	5.0	115	7.4
d <sub>3</sub> Other question	3	0.4	10	1.3	13	0.8
d <sub>4</sub> Confirming answer	14	1.8	40	5.3	54	3.5
d <sub>5</sub> Disconfirm. answer	3	0.4	21	2.8	24	1.5
d <sub>6</sub> Other answer	2	0.3	23	3.0	25	1.6
d <sub>7</sub> Agreeing statement	75	9.5	190	25.0	265	17.1
d <sub>8</sub> Disagree. statement	2	0.3	10	1.3	12	0.8
d <sub>9</sub> Informing statement	<b>391</b>	<b>49.5</b>	<b>305</b>	<b>40.1</b>	<b>696</b>	<b>44.9</b>
d <sub>10</sub> Other	40	5.1	61	8.0	101	6.5
e <sub>1</sub> Test understanding	56	7.1	0	0.0	56	3.6
e <sub>2</sub> Test prior knowledge	111	14.1	1	0.1	112	7.2
e <sub>3</sub> Provide explanation	<b>409</b>	<b>51.8</b>	<b>270</b>	<b>35.5</b>	<b>679</b>	<b>43.8</b>
e <sub>4</sub> Request explanation	47	5.9	95	12.5	142	9.2
e <sub>5</sub> Signal understanding	37	4.7	104	13.7	141	9.1
e <sub>6</sub> Signal non-underst.	1	0.1	16	2.1	17	1.1
e <sub>7</sub> Provide feedback	61	7.7	224	29.5	285	18.4
e <sub>8</sub> Provide assessment	10	1.3	1	0.1	11	0.7
e <sub>9</sub> Provide extra info	26	3.3	22	2.9	48	3.1
e <sub>10</sub> Other	32	4.1	27	3.6	59	3.8
$\Sigma$	790	100.0	760	100.0	1550	100.0

Table 2: Corpus distribution of annotated topics (t<sub>1</sub>–t<sub>4</sub>), dialogue acts (d<sub>1</sub>–d<sub>10</sub>), and explanation moves (e<sub>1</sub>–e<sub>10</sub>) separately for explainer and explainee turns and in total. Per type, the highest value in a column is marked bold.

For dialogue acts (d<sub>1</sub>–d<sub>10</sub>), we see that, quite intuitively, *informing statements* (44.9%) are dominant in explaining dialogues on both sides (explainer 49.5%, explainee 40.1%). However, also *agreeing statements* (17.1%) as well as *check questions* (15.8%) play an important role. The low frequency of *other questions* (0.8%) and *other* (6.5%) suggests that the selected set of dialogue acts cover well what happens in the given kind of dialogues, even though our annotators identified sum acts, such as *disagreeing statements* (0.8%), rarely only.<sup>10</sup>

Similar holds for the explanation moves (e<sub>1</sub>–e<sub>10</sub>): only 3.8% of all 1550 turns belong to *other*.<sup>11</sup> As expected, the core of explaining is to *provide explanations* (43.8%), also explainees do so in 270 turns (35.5%). Besides, they often *provide feedback* (29.5%). Explainers rather *test prior knowledge* (14.1%) and *test understanding* often (7.1%), but also provide feedback sometimes (7.7%).

<sup>10</sup>Notable examples of other dialogue acts the annotators observed include *greetings* (e.g., “Hi, are you Bella?”), *casual chat* (“What do you do?”), and *gratitude* (“Thank you.”).

<sup>11</sup>Here, other cases include *inquiry* (“Hi, are you Bella”) and *introduction* (“Bella, I’m George, nice to meet you.”).

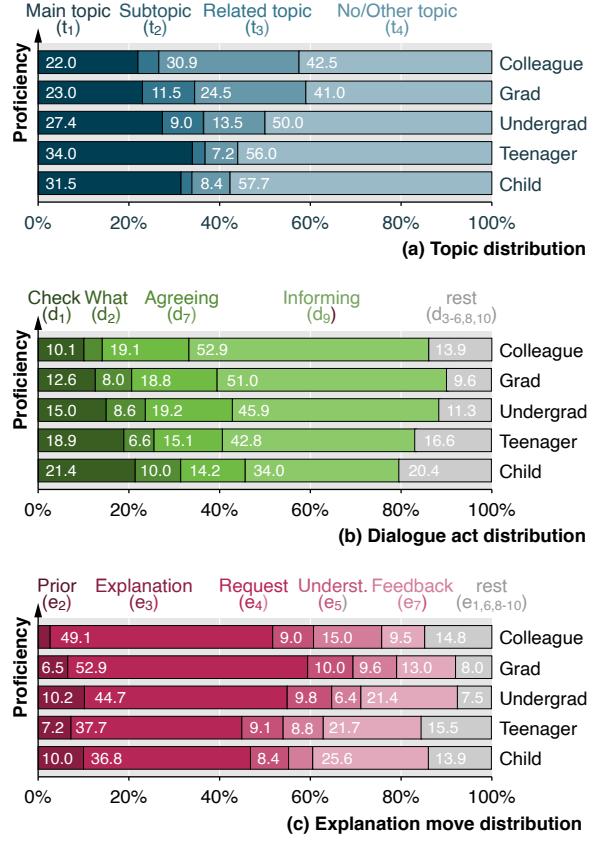


Figure 2: Distribution of topic, discourse act, and explanation act annotations in the corpus, depending on the proficiency of the explainee (from *Child* to *Colleague*).

## 4 Analysis

One main goal of the presented corpus is to learn how humans explain in dialogical settings. This section analyzes commonalities and differences regarding meta-information available in the corpus.

### 4.1 Explaining across Proficiency Levels

First, we explore to what extent explaining differs depending on the proficiency of the explainee. Figure 2 shows the distributions of the three annotated dimensions separately for the five given explainee levels. For dialogue acts and explanation moves, we distinguish only the most frequent labels and merge all others into a class *rest*.

With respect to topic, we see that particularly the discussion of *related topics* grows notably with the explainee’s proficiency, from 8.4% of all annotations for children to 30.9% for colleagues. Conversely, the *main topic* is mentioned less in dialogues with more proficient explainees; the same holds for *no/other topic*. *Subtopics* are considered mainly with grads (11.5%) and undergrads (9.0%), possibly related to the way they learn.

Topic Sequences	Explainer	Explainee	Total
(Main, Rel, Main)	<b>24.6%</b>	7.7%	<b>15.4%</b>
(Main, Rel, Main, Rel, Main)	—	—	7.7%
(Main)	12.3%	<b>18.5%</b>	6.2%
(Rel, Main, Rel, Main, Rel, Main)	—	—	4.6%
(Main, Rel)	3.1%	10.8%	4.6%
(Rel, Main, Rel, Main)	3.1%	—	3.1%
(Main, Sub, Main)	—	—	3.1%
(Main, Sub, Main, Rel, Main)	4.6%	3.1%	3.1%

Table 3: Relative frequencies of all recurring sequences of *main*, *sub*, and *related* topic in the corpus’ dialogues and in the explainers and explainees’ parts alone.

For dialogue acts, the key difference lies between the proportion of *informing statements* and the number of questions asked ( $d_1$  and  $d_2$ ). Whereas the former monotonously goes up from 34.0% (child) to 52.9% (colleague), particularly the use of *check questions* is correlated inversely with proficiency, used mainly to test prior knowledge and to check understanding. A similar behavior can be observed for explanation moves. There, *providing feedback* shrinks from 25.6% to 9.5%, while *providing explanations* mostly grows, with peak at grads (52.9%). In contrast, how often people *request explanations* remains stable across proficiency levels.

## 4.2 Interactions of Topics, Moves, and Acts

Interactions of the annotated dimensions happen between the turns and within a turn. We analyze one example of each here, and, due the limited data size, we look at topics separately from dialogue act and explanation moves.

Inspired by the flow model of Wachsmuth and Stein (2017), Table 3 shows all eight sequences of topics that occur more than once among the 65 dialogues. Each sequence shows the ordering of topics being discussed, irrespective of how often each topic is mentioned in a row. Most dialogues start and end with the main topic, often in alternation with related topics, such as (*Main, Rel, Main*) in 15.4% of all cases (sometimes also with subtopics). The ordering of what *explainers* talk about is similar, whereas *explainees* often focus on the main topic only (18.5%).

Table 4 lists the top-10 pairs of acts and moves. *Informing statements* that *provide explanations* are most common across both explainers (45.9%) and explainees (31.3%). *Agreeing statements* ( $d_7$ ) and *check questions* ( $d_1$ ) cooccur with multiple moves, and especially *providing feedback* happens via different dialogue acts. As expected in the given set-

Labels	Act/Move Pair	Explainer	Explainee	Total
$d_9/e_3$	Informing/Explanation	<b>45.9%</b>	<b>31.3%</b>	<b>38.8%</b>
$d_7/e_7$	Agreeing/Feedback	3.9%	14.2%	9.0%
$d_7/e_5$	Agreeing/Understanding	3.5%	9.1%	6.3%
$d_1/e_2$	Check/Prior	10.5%	—	5.4%
$d_1/e_4$	Check/Request	2.7%	6.8%	4.7%
$d_2/e_4$	What/Request	3.0%	4.5%	3.7%
$d_{10}/e_{10}$	Other/Other	2.8%	2.6%	2.7%
$d_1/e_1$	Check/Understanding	5.1%	—	2.6%
$d_4/e_7$	Confirming/Feedback	1.4%	3.7%	2.5%
$d_9/e_7$	Informing/Feedback	0.5%	4.2%	2.3%

Table 4: Relative frequencies of the ten most frequent pairs of dialogue act and explanation move in the corpus and the differences for explainers and explainees.

Explainer			Explainee		
Word	Frequency	Ratio	Word	Frequency	Ratio
here	0.16%	4.20	yes	0.21%	5.12
around	0.12%	4.03	mean	0.14%	4.20
space	0.24%	3.32	stuff	0.11%	3.11
light	0.18%	2.96	oh	0.16%	2.75
earth	0.10%	2.65	yeah	0.65%	2.70
us	0.15%	2.39	many	0.12%	2.39
want	0.14%	2.28	interesting	0.12%	2.11
going	0.22%	2.19	well	0.21%	1.94
point	0.11%	2.11	like	1.10%	1.85
thing	0.18%	1.93	no	0.18%	1.83

Table 5: The top-10 words used specifically by explainers and explainees, respectively, along with the relative frequency (minimum 0.1%) and specificity ratio (e.g., explainees say “yes” 5.12 times as often as explainers).

ting, explainees never check for prior knowledge or understanding ( $d_1/e_2$ ,  $d_1/e_1$ ). Instead, they agree by providing feedback or signaling understanding ( $d_7/e_7$ ,  $d_7/e_5$ ) much more often than explainers.

## 4.3 Language of Explainers and Explainees

Finally, we investigate basic differences in the language of the two sides: We determine the words that are often used by explainers (at least 0.1% of all words) and rarely by explainees, or vice versa.

Table 5 presents the 10 most specific words on each side. Aside from some topic-specific words (e.g., “light”), the explainer’s list includes typical words used in meta-language, as in this explanation to a teenager: “I want to know if you agree, sleep is the coolest *thing* you’ve ever heard of.” On the explainees’ side, we find multiple reactive words, such as “oh” and “interesting”, but also indicators of vagueness, as in this colleague’s response to an explanation of hacking: “So all kind of older logic and *stuff like* that. So, I mean, it’s sort of based on, *like*, you’re presented the little MUX chip.”

## 5 Experiments

The second goal of the corpus is to serve the creation of XAI systems that mimic human explainers. As an initial endeavor, this section reports on baseline experiments on the computational prediction of topics, dialogue acts, and explanation moves.

### 5.1 Experimental Setup

We evaluate three models based on BERT (Devlin et al., 2019), along with a simple majority baseline, for predicting each dialogue turn dimension in 13-fold cross-topic validation: For each main topic, we trained one model on the other 12 topics and tested it against the labels of the respective dimension. We average the resulting  $F_1$ -scores over all 13 folds.<sup>12</sup> Figure 3 illustrates the three BERT variants.

**BERT-basic** The first model simply adds a classification head to BERT. It takes as input the dialogue’s main topic and the turn’s text,  $x_i$  (separated by [SEP]), as well as the label  $y_i$  to predict (topic  $t_i$ , dialogue act  $d_i$ , or explanation move  $e_i$ ). We trained the model for five epochs, optimizing its  $F_1$ -score on the turns of two main topics. We balanced the training set using oversampling to prevent the model from only predicting the majority label.

**BERT-sequence** Turns made in explaining dialogues depend on previous turns, for example, a conclusion on the *main topic* may be preceded by a *related topic* (see Table 3). In the second model, we exploit such dependencies with turn-level sequence labeling: Given the sequence  $(x_1, \dots, x_n)$  of all turns in a dialogue, the input to predicting a label  $y_i$  of  $x_i$  is the turn’s history  $(x_1, \dots, x_{i-1})$  along with all previously predicted labels  $(y_1, \dots, y_{i-1})$  of the same dimension. For each turn, we encode the history in a `CLS` embedding with BERT. Then, we pass all labels and `CLS` embeddings through a CRF layer to model the label’s dependencies.

**BERT-multitask** Finally, the interaction of topic  $t_i$ , act  $d_i$ , and move  $e_i$  in a turn may be relevant. For example, an *informing statement* likely provides an *explanation* (see Table 4). Our third model thus learns to classify all three dimensions jointly in a multitask fashion, based on multitask-NLP.<sup>13</sup> We trained one multitask model each with one of the three dimensions as main task and the others as

<sup>12</sup>All models start from the `bert-based-uncased`, and are trained with a learning rate of  $2e^{-5}$  and a batch size of 4.

<sup>13</sup>Multitask NLP, <https://multi-task-nlp.readthedocs.io>

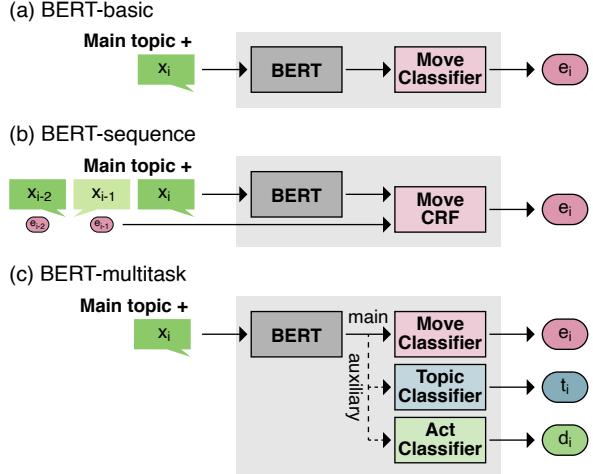


Figure 3: Sketch of the three evaluated models, here for predicting a turn’s explanation move,  $e_i$ : (a) *BERT-basic* labels a turn in isolation. (b) *BERT-sequence* takes the labels of previous turns into account. (c) *BERT-multitask* classifies all three turn dimensions simultaneously.

Approach	Main T. ( $t_1$ )	Sub- T. ( $t_2$ )	Related T. ( $t_3$ )	No/Oth. T. ( $t_4$ )	Macro F <sub>1</sub> -Score
BERT-basic	0.58	0.11	<b>0.44</b>	<b>0.89</b>	0.51
BERT-sequence	<b>0.61</b>	<b>0.13</b>	<b>0.44</b>	<b>0.89</b>	<b>0.52</b>
BERT-multitask	0.43	0.04	0.36	0.81	0.41
Majority baseline	0.00	0.00	0.00	0.66	0.17

Table 6: Topic prediction results: The  $F_1$ -scores of the evaluated BERT models for each considered relation to the main topic,  $t_1-t_4$ , as well as the macro-averaged  $F_1$ -score. The best value in each column is marked bold.

auxiliary tasks, oversampling with respect to the main task. To this end, we employ a shared BERT encoder and three classification heads, one for each task. The final loss is the weighted average of the three classification losses, with weight 0.5 for the main task and 0.25 for both others. We trained the models for 10 epochs allowing them to converge.

### 5.2 Results

Tables 6–8 show the individual and the macro  $F_1$ -scores for all three dimensions.

*BERT-sequence* performs best across all three labeling tasks, highlighting the impact of modeling the sequential interaction in dialogues. It achieves a macro  $F_1$ -score of 0.52 for topics, 0.47 for dialogue acts, and 0.43 for explanation moves. However, likely due to data sparsity, some labels remain hard to predict, such as *Subtopic* ( $t_2$ ), *disagreement statements* ( $d_8$ ), and *provide assessment* ( $e_8$ ).

*BERT-basic* beats *BERT-sequence* on a few la-

Approach	Check Q. (d <sub>1</sub> )	What/H. Q. (d <sub>2</sub> )	Other Q. (d <sub>3</sub> )	Confirm. A. (d <sub>4</sub> )	Disconf. A. (d <sub>5</sub> )	Other A. (d <sub>6</sub> )	Agree. St. (d <sub>7</sub> )	Disagr. St. (d <sub>8</sub> )	Inform. St. (d <sub>9</sub> )	Other (d <sub>10</sub> )	Macro F <sub>1</sub> -Score
BERT-basic	<b>0.76</b>	<b>0.73</b>	0.00	0.33	<b>0.67</b>	0.00	0.51	0.00	<b>0.87</b>	0.57	0.44
BERT-sequence	<b>0.76</b>	0.72	0.00	<b>0.35</b>	<b>0.67</b>	0.00	** <b>0.69</b>	0.00	<b>0.87</b>	<b>0.61</b>	<b>0.47</b>
BERT-multitask	0.54	0.49	0.00	0.29	0.59	0.00	0.53	<b>0.09</b>	0.84	0.44	0.38
Majority baseline	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.00	0.06

Table 7: Dialogue act prediction results: The F<sub>1</sub>-scores of the evaluated BERT models for each considered dialogue act, d<sub>1</sub>–d<sub>10</sub>, as well as the macro-averaged F<sub>1</sub>-score. The best value in each column is marked bold.

Approach	Test U. (e <sub>1</sub> )	Test P.K. (e <sub>2</sub> )	Provide Ex. (e <sub>3</sub> )	Request Ex. (e <sub>4</sub> )	Signal U. (e <sub>5</sub> )	Signal N.U. (e <sub>6</sub> )	Provide Fe. (e <sub>7</sub> )	Provide As. (e <sub>8</sub> )	Provide E.I. (e <sub>9</sub> )	Other (e <sub>10</sub> )	Macro F <sub>1</sub> -Score
BERT-basic	<b>0.27</b>	<b>0.64</b>	<b>0.84</b>	0.60	0.29	<b>0.34</b>	0.51	0.00	<b>0.11</b>	0.50	0.41
BERT-sequence	<b>0.27</b>	<b>0.64</b>	<b>0.84</b>	<b>0.64</b>	<b>0.33</b>	0.21	** <b>0.60</b>	<b>0.15</b>	0.08	<b>0.56</b>	<b>0.43</b>
BERT-multitask	0.21	0.54	0.80	0.40	0.16	0.32	0.53	0.00	0.08	0.35	0.34
Majority baseline	0.00	0.00	0.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06

Table 8: Explanation move prediction results: The F<sub>1</sub>-scores of the evaluated BERT models for each considered explanation move, e<sub>1</sub>–e<sub>10</sub>, as well as the macro-averaged F<sub>1</sub>-score. The best value in each column is marked bold.

bels, such as *signal non-understanding* (e<sub>8</sub>), but cannot compete overall. *BERT-multitask* performs worst among the three models. We attribute this to the data imbalance: While oversampling helps with respect to the main task, it does not benefit the label distribution of the auxiliary tasks. Also, optimizing the loss weights of the three tasks may further aid multitask learning, but such an engineering of prediction models is not the focus of this work.

## 6 Conclusion

How humans explain in dialogical settings is still understudied. This paper has presented a first corpus for computational research on controlled explaining dialogues, manually annotated for topics, dialogue acts, and explanation moves. Our analysis has revealed intuitive differences in the language of explainers and explainees and their dependence on the explainee’s proficiency. Moreover, baseline experiments suggest that a prediction of the annotated dimensions is feasible and benefits from modeling interactions. With these results, we lay the ground towards more human-centered XAI. We expect that respective systems need to learn to how to explain depending on the explainee’s reactions, and how to proactively lead an explaining dialogue to achieve understanding on the explainee’s side.

A limitation of the corpus lies in the restricted corpus size caused by the availability of source data, preventing deeper statistical analyses and likely rendering a direct training of dialogue systems on the corpus hard. Also, it remains to be explored what findings generalize beyond the controlled setting of

the given dialogues. Future work should thus target both the scale and the heterogeneity of explaining data, in order to provide the pervasive communicative process of explaining the attention it deserves.

## Acknowledgments

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), partially under project number TRR 318/1 2021 – 438445824 and partially under SFB 901/3 – 160364472. We thank Meisam Booshehri, Henrik Buschmeier, Philipp Cimiano, Josephine Fisher, Angela Grimminger, and Erick Ronoh for their input and feedback to the annotation scheme. We also thank Akshit Bhatia for his help with the corpus preparation as well as the anonymous freelancers on Upwork for their annotations.

## 7 Ethical Statement

We do not see any immediate ethical concerns with respect to the research in this paper. The data included in the corpus is freely available. All participants involved in the given dialogues gave their consent to be recorded and received expense allowances, as far as perceivable from the Wired web resources. As discussed in the paper, the three freelancers in our annotation study were paid about \$13 per hour, which exceeds the minimum wage in most US states and is also conform to the standards in the regions of our host institution. In our view, the provided prediction models target dimensions of dialogue turns that are not prone to be misused

for ethically doubtful applications.

## References

- Khalid Al Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. 2016. [A news editorial corpus for mining argumentation strategies](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3433–3443. The COLING 2016 Organizing Committee.
- Khalid Al Khatib, Henning Wachsmuth, Kevin Lang, Jakob Herpel, Matthias Hagen, and Benno Stein. 2018. [Modeling deliberative argumentation strategies on wikipedia](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2545–2555. Association for Computational Linguistics.
- Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. [Explainable artificial intelligence \(XAI\): Concepts, taxonomies, opportunities and challenges toward responsible AI](#). *Information Fusion*, 58:82–115.
- Sarah Bourse and Patrick Saint-Dizier. 2012. [A repository of rules and lexical resources for discourse structure analysis: the case of explanation structures](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2778–2785, Istanbul, Turkey. European Languages Resources Association (ELRA).
- Harry Bunt, Jan Alexandersson, Jean Carletta, Jaewoong Choe, Alex Chengyu Fang, Koiti Hasida, Kiyong Lee, Volha Petukhova, Andrei Popescu-Belis, Laurent Romary, Claudia Soria, and David Traum. 2010. [Towards an ISO standard for dialogue act annotation](#). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).
- Roberto Confalonieri, Tarek R. Besold, Tillman Weyde, Kathleen Creel, Tania Lombrozo, Shane T. Mueller, and Patrick Shafto. 2019. [What makes a good explanation? Cognitive dimensions of explaining intelligent machines](#). In *Proceedings of the 41th Annual Meeting of the Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation, Montreal, Canada, July 24-27, 2019*, pages 25–26.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Andrei Dulceanu, Thang Le Dinh, Walter Chang, Trung Bui, Doo Soon Kim, Manh Chien Vu, and Seokhwan Kim. 2018. [PhotoshopQuiA: A corpus of non-factoid questions and answers for why-question answering](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Myroslava O. Dzikovska, Rodney D. Nielsen, and Chris Brew. 2012. [Towards effective tutorial feedback for explanation questions: A dataset and baselines](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 200–210, Montréal, Canada. Association for Computational Linguistics.
- Angela Fan, Yacine Jernite, Ethan Perez, David Granger, Jason Weston, and Michael Auli. 2019. [ELI5: Long form question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.
- Josefine Finke, Ilona Horwath, Tobias Matzner, and Christian Schulz. 2022. [\(de\)coding social practice in the field of xai: Towards a co-constructive framework of explanations and understanding between lay users and algorithmic systems](#). In *Artificial Intelligence in HCI*, pages 149–160, Cham. Springer International Publishing.
- Lionel Fontan and Patrick Saint-Dizier. 2008. [Analyzing the explanation structure of procedural texts: Dealing with advice and warnings](#). In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 115–127. College Publications.
- Daniel Fried, Jacob Andreas, and Dan Klein. 2018. [Unified pragmatic models for generating and following instructions](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1951–1963, New Orleans, Louisiana. Association for Computational Linguistics.
- Alan Garfinkel. 2009. *Forms of Explanation: Rethinking the Questions in Social Theory*, revised edition. Yale University Press, New Haven & London, New Haven; London.
- Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. [Explaining explanations: An overview of interpretability of machine learning](#). ArXiv: 1806.00069.
- Bryce Goodman and Seth Flaxman. 2017. [European union regulations on algorithmic decision-making and a “right to explanation”](#). *AI Magazine*, 38(3):50–57.

- Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. **The argument reasoning comprehension task: Identification and reconstruction of implicit warrants.** In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1930–1940. Association for Computational Linguistics.
- Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. **Learning whom to trust with MACE.** In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia. Association for Computational Linguistics.
- Peter Jansen, Niranjan Balasubramanian, Mihai Surdeanu, and Peter Clark. 2016. **What’s in an explanation? characterizing knowledge and inference requirements for elementary science exams.** In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2956–2965, Osaka, Japan. The COLING 2016 Organizing Committee.
- Pamela W. Jordan, Maxim Makatchev, and Umarani Pappuswamy. 2006. **Understanding complex natural language explanations in tutorial applications.** In *Proceedings of the Third Workshop on Scalable Natural Language Understanding*, pages 17–24, New York City, New York. Association for Computational Linguistics.
- Lei Li, Yongfeng Zhang, and Li Chen. 2021. **Personalized transformer for explainable recommendation.** In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4947–4957, Online. Association for Computational Linguistics.
- William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- Tim Miller. 2019. **Explanation in artificial intelligence: Insights from the social sciences.** *Artificial Intelligence*, 267:1–38.
- Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. **SemEval-2017 task 3: Community question answering.** In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48, Vancouver, Canada. Association for Computational Linguistics.
- Katharina J. Rohlfing, Philipp Cimiano, Ingrid Scharlau, Tobias Matzner, Heike M. Buhl, Hendrik Buschmeier, Elena Esposito, Angela Grimmlinger, Barbara Hammer, Reinhold Häb-Umbach, Ilona Horwath, Eyke Hüllermeier, Friederike Kern, Stefan Kopp, Kirsten Thommes, Axel-Cyrille Ngonga Ngomo, Carsten Schulte, Henning Wachsmuth, Petra Wagner, and Britta Wrede. 2021. **Explanation as a social practice: Toward a conceptual framework for the social design of ai systems.** *IEEE Transactions on Cognitive and Developmental Systems*, 13(3):717–728.
- Xuelin Situ, Ingrid Zukerman, Cecile Paris, Sameen Maruf, and Gholamreza Haffari. 2021. **Learning to explain: Generating stable explanations fast.** In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5340–5355, Online. Association for Computational Linguistics.
- Youngseo Son, Nipun Bayas, and H. Andrew Schwartz. 2018. **Causal explanation analysis on social media.** In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3350–3359, Brussels, Belgium. Association for Computational Linguistics.
- Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. **Dialogue act modeling for automatic tagging and recognition of conversational speech.** *Computational Linguistics*, 26(3):339–374.
- John M. Swales. 1990. *Genre Analysis: English in Academic and Research Settings*. Cambridge University Press.
- Keith Vander Linden. 1992. The expression of local rhetorical relations in instructional text. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 318–320.
- Henning Wachsmuth and Benno Stein. 2017. **A universal model for discourse-level argumentation analysis.** *Special Section of the ACM Transactions on Internet Technology: Argumentation in Social Media*, 17(3):28:1–28:24.
- Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. **RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes.** In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1368, Brussels, Belgium. Association for Computational Linguistics.
- Ziqi Zhang, Philip Webster, Victoria Uren, Andrea Varga, and Fabio Ciravegna. 2012. **Automatically extracting procedural knowledge from instructional texts using natural language processing.** In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 520–527, Istanbul, Turkey. European Language Resources Association (ELRA).

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/355874213>

# A qualitative mapping of Darkweb marketplaces

Conference Paper · December 2021

DOI: 10.1109/eCrime54498.2021.9738766

---

CITATIONS

5

READS

1,212

4 authors, including:



Morten Falch

Aalborg University

121 PUBLICATIONS 741 CITATIONS

[SEE PROFILE](#)



Emmanouil Vasilomanolakis

Technical University of Denmark

66 PUBLICATIONS 986 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



HosTaGe [View project](#)



Blockchain security [View project](#)

# A qualitative mapping of Darkweb marketplaces

Dimitrios Georgoulias  
*Cyber Security Group*  
*Aalborg University*  
Copenhagen, Denmark  
dge@es.aau.dk

Jens Myrup Pedersen  
*Cyber Security Group*  
*Aalborg University*  
Copenhagen, Denmark  
jens@es.aau.dk

Morten Falch  
*Cyber Security Group*  
*Aalborg University*  
Copenhagen, Denmark  
falch@es.aau.dk

Emmanouil Vasilomanolakis  
*Cyber Security Group*  
*Aalborg University*  
Copenhagen, Denmark  
emv@es.aau.dk

**Abstract**—Darkweb marketplaces have evolved greatly since the rise of the Silk Road in 2011, the first platform of its kind, and have become a highly profitable underground trading ecosystem, which provides anonymity for both buyers and sellers. Law enforcement along with researchers, have been successful in taking down marketplaces over the years. However, the combination of mechanisms implemented by these platforms (e.g. payment mechanisms, cryptocurrencies, trust systems), along with the success of the Tor network’s anonymity properties, have made marketplaces much more enticing to users, while providing ease of access and use, as well as resilience against hostile actions. Through qualitative methods, this paper presents a mapping of darkweb marketplaces. We systematically investigate the operation of 41 marketplaces, along with 35 vendor shops, and gather information about the mechanisms and features implemented. Additionally, to acquire real world information, we explore the marketplaces’ integrated forums, as well as 3 popular independent ones, focusing on discussions between vendors, buyers and marketplace owners, on topics related to illegal trading. We believe that gaining an up-to-date and deep understanding of the framework that marketplaces are built upon, is the first step towards discovering weak spots in the cyber security product and service market, with the disruption of its operation being the ultimate goal.

**Index Terms**—darkweb, marketplaces, illegal trading, cryptocurrency, cybercrime

## I. INTRODUCTION

The year 2010 marks the appearance of underground marketplaces in the Darkweb. It all started with the *The Farmer’s Market*, which moved its operation from the clearweb to the Tor network. However, *Silk Road* is considered as the first successful darkweb marketplace of its type, due to its much greater impact [1]. This type of marketplace could effectively provide anonymity to its clients. This was achieved through utilizing the Tor network, and specifically its hidden service function. Potential buyers would use the hidden service’s onion address to access the marketplace, remaining anonymous while doing so. They would then be met with a variety of vendors offering products and services, from which they could choose according to their personal preference. Furthermore, the implementation of Bitcoin (BTC) transactions, certainly added to the anonymity of all involved parties, namely the buyers, sellers, and marketplace owners.

This adoption of an online marketplace, has served as a blueprint for all the marketplaces that succeeded Silk Road in the last decade. Implementations have only become more robust and resilient against takedown and infiltration attempts from LEAs. Furthermore, the variety of products and services available for purchase, has increased considerably, along with their availability and the cryptocurrencies that can be used to acquire them. Darkweb marketplaces are part of an ecosystem that operates similarly to legitimate enterprises, with the most important addition being anonymity. They present mechanisms, such as vendor reputation systems, escrow, communication encryption (e.g. PGP), review systems, integrated forum sections with discussions, and customer support functions, all of which aim to build a chain of trust between the buyers, sellers and the marketplace owners. Furthermore, this trust is achieved without either of the parties involved, revealing their identities to one another. The darkweb is considered somewhat of a mystery by most users, which creates hesitation, mistrust and even fear, disheartening users from ever using it. Consequently, minimizing the risk of clients getting scammed by vendors, in combination with anonymous transactions and communications, as well as the sense of a community through forum discussions (both integrated and independent), create an environment where clients can feel safe and more encouraged to carry out purchases.

The products and services available on the darkweb marketplaces present great variety. Some popular examples are drugs, guns, bank card and account credentials, social network platform accounts (e.g. hacked Facebook and Twitter accounts), counterfeits (e.g. fake driving licenses), hacking services, exploit kits, botnet services (e.g. DDoS attacks, botnet rentals and sales) and malware. However, since the COVID-19 epidemic outbreak, the marketplace scene has adapted. Many vendors have been trying to capitalize on people’s fear of infection, and the global need for protection against the virus. This has lead to marketplace product listings also including testing kits, vaccines, forged test results, as well as fake vaccination certificates [2].

Darkweb marketplaces have been getting more and more successful over the years. The revenue generated reached approximately \$1.7 billion in 2020, 75% (\$1.3 billion) [3] of which was reportedly generated by the Russian marketplace Hydra, making it by far the most profitable marketplace. Furthermore, for the year 2020, ranking countries according to

both the value sent to these marketplaces (purchases) and the value earned by them (revenue), presents Russia dominating the top of the list in both aspects, with the United States and Ukraine occupying the second and third place respectively [3].

In this paper, we investigate the current state of marketplaces in the darkweb. We focus on 41 marketplaces and their forums, but we also navigated through 35 vendor shops, as well as 3 independent darkweb forums, in order to gain a deeper understanding of the entire darkweb ecosystem. Our contribution lies in mapping the darkweb marketplace infrastructure, by documenting the mechanisms and features implemented by marketplaces in the darkweb, as well as the practices applied by vendors, buyers and marketplace owners. We argue that gaining detailed insights on the infrastructure's different characteristics and properties, is a stepping stone towards vulnerability discovery, exploitation, and consequently, the disruption of darkweb operations related to cyber attack products and services, such as botnets, malware and exploit trading.

## II. METHODOLOGY

The information gathered for the purposes of this paper, originate from 3 main sources; 41 marketplaces, including their integrated forums, 35 vendor shops, and 3 popular darkweb forums. Regarding the choice of platforms, the marketplaces we include are all that were operational at the time of this paper (August 2021), and the forums were chosen based on popularity. Vendor shops, being shops of individual sellers, present very limited variety of features and properties, and do not provide as much insight as marketplaces, since they are considerably smaller. However, navigating through them provided additional data on various basic mechanisms that are shared in common with the bigger marketplaces, but since their number is quite high, for the purpose of this paper we deemed exploring 35 of them to be sufficient. The information we document originates from a combination of Frequently Asked Questions (FAQ) sections, as well as guides and discussions between marketplace users, both vendors and buyers, found on the forums. Visiting each marketplace and vendor shop individually, and attempting to test out each platform's features and infrastructure, was necessary towards gaining as much insight as possible on the darkweb market.

In more detail, the way this process was executed, was firstly visiting the marketplace, and documenting the CAPTCHA mechanisms. We then proceed to make a user account, since in the majority of the platforms it is a requirement (apart from some special cases), to gain access to the product listings. In many cases there would be another CAPTCHA required to finalize the registration, which was also documented. The next step included navigating though the FAQ section and forum sections of the site, where we would typically acquire information on the features and properties of each marketplace. We would then focus on browsing through several product listings, vendor profiles, and user reviews, gaining insight on elements such as currency, payment methods, and reputation

systems. Furthermore, we would also test the features discovered, along with some basic mechanisms such as deposits and withdrawals, as well as go through the purchase process up until the point of payment. However, we did not carry out any purchases due to ethical and legal considerations. The way we tracked down the onion addresses for all of the platforms we visited, was through *introduction points*, sites (often both on the clearweb and darkweb) serving as directories for hidden services. Apart from the procedure described, previous academic research on some of the elements mentioned in this paper, also provided guidance, contributing to our efforts (see Section IV).

### A. Ethical issues

At this point we need to address the ethical standpoint of this paper. All of the platforms we investigated are part of the public digital space and free to access. Since we interacted with each platform's functions as plain users, we did not cause any disruption to the services, and did not in any way negatively affect the experience of other users. Furthermore, we did not acquire or analyze any type of user sensitive data. We only utilized publicly available sources, such as forum discussions and reviews, without disclosing information that could potentially breach the privacy of any individuals or risk exposing their identity.

It has been argued by previous research that these platforms can be viewed as a safer alternative to conventional real-world drug trading, due to its digital nature [4]–[9]. Hence, it should be clearly stated that the goal of this work is not to bring down marketplaces. This research instead aims to be used as a stepping stone towards disrupting specific cyber attack services, with Distributed Denial of Service (DDoS) service providers as a prime example [10]. Lastly, we do not provide a full list of the targeted marketplaces, but we do however mention some of them by name throughout this article, in order to showcase various operational features and example mechanisms that they implement. The rationalization behind this is that we want to avoid directing traffic to as many platforms as possible, but without hindering the scientific contribution of this paper.

## III. MARKETPLACE ELEMENTS

In the effort of mapping darkweb marketplaces, we categorize the properties of these platforms into *Access & Authentication*, *Products & Purchases*, *Shipping & Delivery*, *Vendor Reputation*, *Support*, *Disputes & Community*, and *Marketplace Revenue*.

### A. Access & Authentication

1) *Access*: The majority of darkweb selling points, as well as all of the 41 marketplaces we investigate in this paper, are free to access and can be located through both clearweb and darkweb websites or using darkweb search engines such as *Torch*. However, there are a number of platforms that are only available through a registration fee, or through invites, which are made available to trusted users. These users can

vouch for newer members, that will then avoid paying for the access, which can get quite expensive (e.g. the *KickAss* forum fee is \$450). Despite the restricted access mechanism, invites for some of these platforms can often be found for sale on marketplaces, sometimes for a fraction of the price. Additionally, it is not uncommon practice for paid-access shops and forums to offer some kind of discount to attract new members, which they will advertise in popular forums such as *Dread*.

**2) Protection Mechanism - CAPTCHAs:** The majority of the 41 marketplaces we visited for the purposes of this article implemented DDoS and crawling protection (see Appendix C). Most platforms would firstly place the user in a queue, lasting a few seconds, and then prompt a CAPTCHA which would either be a standalone mechanism, or part of the registration/login page. In the former scenario, after solving the first CAPTCHA, there would usually be a second one embedded in the registration/login page. The CAPTCHAs implemented are typically text-based, image-based, e.g. image puzzle solving or image matching under a specific context, in a question and answer format, e.g. mathematical equations, and lastly in an analog clock format (see Figures 1 4 and Appendix D<sup>1</sup>). In this case the user is met with an analog clock face showing a random hour/minute combination. They then have to beat a one minute timer, which starts counting down immediately after the web page loads, in which time they have to choose the two correct numbers corresponding to the hours and minutes of the time shown, in a 12 hour format, from two drop down menus located below the clock itself. The *Vice City* marketplace also uses a CAPTCHA where the user is given a set of 9 symbols, some of which are colored in, along with a 3x3 table with empty circles. To solve the CAPTCHA, the user must then choose the circles that share the same position on the table, as the colored symbols on the given image. The *ASAP* marketplace, uses a set of moving text characters, the user must distinguish and input. Furthermore, the *Yakuza Market* CAPTCHA implementation is the solution of a basic mathematical equation, while the *Nemesis* marketplace utilizes an image based puzzle, where a photo is split into 24 blocks, with 5 of them not matching. The user needs to simply choose the misplaced image blocks. The *Monopoly* marketplace CAPTCHA, out of a set of rings, requires the user to click on the broken ring, while the *Kingdom Market*, deploys an image-based numerical puzzle, where the user needs to click on 9 boxes containing numbers, in the correct ascending order. The *Majestic Garden* market/forum, prompts the user with a text based CAPTCHA, along with 3 simple questions/puzzles. *CannaHome* after a simple text based CAPTCHA, deploys a secondary mechanism, where some characters of a small text are marked with red arrows. The user needs to pick out these characters, input them in the bracket below and they can then proceed to the homepage. Lastly, it should be noted that all of the CAPTCHAs with a

timer would be standalone mechanisms, and in most of these cases there would be a second CAPTCHA at the login page.

**3) Marketplace Verification:** An optional, but crucial step in regard to the user's security, is the verification of the marketplace's identity (see Appendix C). In the darkweb, it is quite common for fake mirror addresses to make their appearance, in an effort to phish users, by imitating the original marketplace. This often occurs in the case of a marketplace's seizure by Law Enforcement Agencies (LEAs). In this case, cybercriminals take advantage of the seizure, and rush to set up a new hidden service, which poses as the original marketplace, where users get phished and scammed. For this reason marketplace owners implement the Pretty Good Privacy (PGP) protocol for authentication. They create a key pair, public and private, and they use the private key to create signed messages, that the users can then verify using the public key. The user can find the public key of the marketplace, on the platform itself (often behind another CAPTCHA), as well as on popular darkweb forums and *introduction points*<sup>2</sup>, which also adds to its validity. One of the two main practical uses of this mechanism, is to authenticate the list of onion addresses that the marketplace can be accessed through, also referred to as mirrors. The marketplace owners create a message which contains all of the mirror addresses, and then sign this message with their private key, proving the legitimacy of the hidden service. This way the user can be certain that they are visiting the original marketplace, by locating the onion address they are using to connect in the signed list of mirror addresses. The second application of the PGP protocol, is to verify the identity of the marketplace owners. This message is often referred to as a *Canary*, it traditionally contains the date and timestamp of its issuing, and it is renewed frequently. In some cases, this message was found to also contain news headlines from popular websites or darkweb forums, proving the message was created recently (e.g. White House Market). Through this system, the users are reassured that the individuals behind the marketplace's operation are still the original owners. In some cases the two aforementioned messages, are combined into a single one, which is yet again updated with a set frequency. It should be noted that the marketplaces will hold onto the same private key, since it essentially is the proof of the marketplace ownership, and serves as the foundation of the entire authentication mechanism. An additional verification method implemented by many marketplaces, is including the onion address of the hidden service, in the background image of the CAPTCHA. This helps the user ascertain that they are not visiting a fake, identical to the original, platform. Lastly, another factor that can contribute towards determining a marketplace's validity, is forum posts of esteemed members, publicly announcing their support towards a platform, as well as discussions providing positive or negative feedback.

**4) Registration:** After going through the queue, and solving the standalone CAPTCHA (if one is utilized), users are able to

<sup>1</sup>We discovered several different CAPTCHAs, but they were a similar implementation to the ones illustrated in Figure 1.

<sup>2</sup>Introduction points are sites, both in the clearweb and in the darkweb, that contain onion addresses of several platforms, often along with their PGP keys. Examples include *Recon* and *Dark.Fail*.

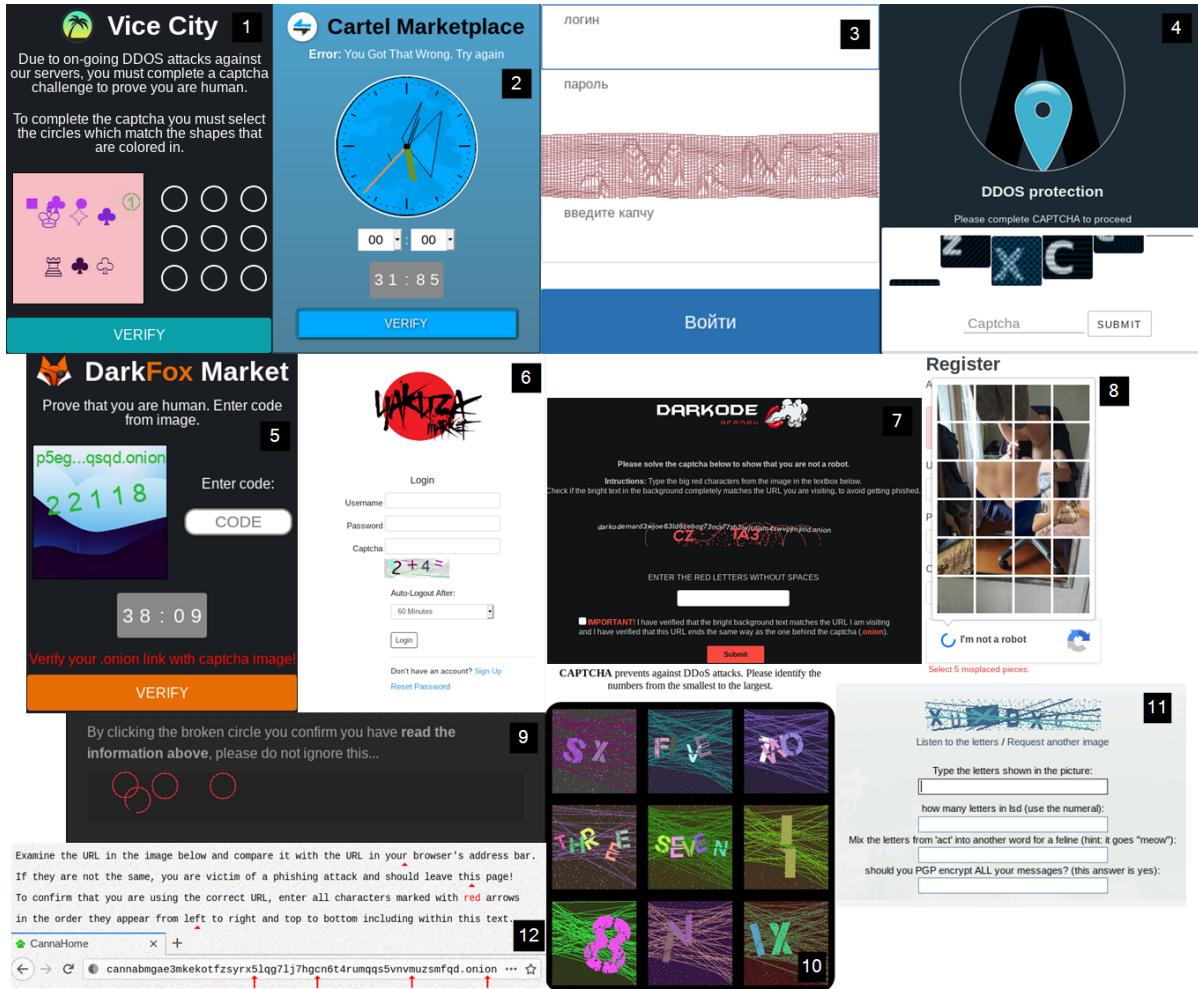


Fig. 1. CAPTCHAs from popular darkweb marketplaces: 1. Vice City, 2. Cartel Marketplace, 3. Hydra, 4. ASAP, 5. DarkFox Market, 6. Yakuza Market, 7. DarkOde Reborn Market, 8. Nemesis Marketplace, 9. Monopoly Marketplace, 10. Kingdom Market, 11. The Majestic Garden, 12. CannaHome

either log in, in the case of an existing account, or register for a new one. In the case of registration, the process is simple. The information the user has to input in the form, are their username and password, and in most cases a 6-digit pin, which serves authentication purposes. Some marketplaces may also have mandatory Two-Factor Authentication (2FA), which in most cases translates into the user entering their PGP key (see Section III-A5). On one particular platform, the user PGP public key was even used instead of a password, for login purposes. In the majority of the marketplaces we investigated, the last step included a mnemonic given to the user, which was either a simple sentence composed of random everyday words, or a string of random characters. Since none of these platforms required an e-mail address in the registration process (apart from some minor exceptions), this mnemonic is to be used in the case the user ever needs to recover their forgotten login credentials. To finalize the registration, the user has to verify their mnemonic, in most cases solve a CAPTCHA, and then they can access the marketplace through the login page. At this point it should be mentioned that in most of the marketplaces,

the user cannot reach the product listings unless they register and login with their account (see Appendix C). However, there was one specific platform that allowed for purchases without the need of registering an account, with the PGP key of the buyer acting as the sole identification method.

5) *User Authentication:* After establishing their account, the user can login using with their credentials, and in most cases, by additionally solving a CAPTCHA. However, users also have the option of setting up 2FA (see Appendix C), which is achieved through PGP or by using a Time-based One-Time Password (TOTP) [11]. In the case of PGP, the user must initially register their public key in their account. Every time they try to log in, after entering their password, the marketplace will use that public key to sent an encrypted message to the user, which contains an additional passphrase. The user must then decrypt the message, derive the passphrase and enter it to complete the log in process. If TOTP is chosen as the authentication method, the user is provided with a QR code, as well as a text code, both intended to be used for generation of one-time codes. This can be accomplished using

authentication applications, such as Google Authenticator or KeePassXC.

### B. Products & Purchases

1) *Product Listings:* The products available in the darkweb marketplaces have been well documented over the years [12]–[14], with more recent work even accounting for the changes that came as a result of the COVID-19 pandemic [15]. For this reason, we decided to mainly focus on the framework that surrounds the listing process of these products, as well as the code of conduct that dictates how they are carried out.

Depending on the platform in question, the rules regarding product listings can slightly vary. Marketplaces will have rules in place forbidding certain products from being listed on the platform. These products are usually child pornography, terrorism related products, weapons, human/animal abuse material, murder for hire services, and most recently, so-called COVID-19 “cures”. These individual types of products and services, can still be found in dedicated vendor shops, with some being more difficult to track down than others, due to their varying level of legality and how closed the corresponding community is (e.g. firearm versus child pornography vendor shops).

In addition to the rules regarding the products and services, vendors must follow certain requirements, in order to create listings on the platform. In some marketplaces these requirements are obligatory, but in others, vendors are given more freedom. Vendors are primarily asked to provide information on the type of their product, exact quantity and price, production origin, an image of the product, the shipping available destinations and origin, as well as shipping methods and their pricing. This applies to physical product listings, since digital product listings (e.g. stolen bank credential information), do not need to include any information related to shipping. Some marketplaces may be very specific regarding this information. For example, *Cartel Marketplace* explicitly asks for an image showing a large quantity of the product, along with a piece of paper stating the names of the vendor and the marketplace.

There is also precedent of marketplaces having listings of various products, but without implementing the typical “add to cart” mechanism, that is used on legitimate platforms on the clearweb. An example is the *Cave Tor* marketplace, which apart from the product information, they will only include the vendors’ contact information, that potential buyers can use to set up the purchase privately with the vendor. Some marketplaces, such as *The Majestic Garden*<sup>3</sup>, will not even display product listings, but will adopt a forum architecture, where clients can find vendors for the products they need in specific sections and threads of the forum.

A big contributor to a vendor’s success on a marketplace is also the level of exposure that their listed products are able to get. Clients visiting a marketplace, will find that some products are being showcased, taking priority over others. This

<sup>3</sup>This specific marketplace is not included in the list of 41 platforms we investigated, since it was not free to access. We did however document its CAPTCHA mechanism (see Figure 1) and found information regarding its operation through forums discussions.

is done through a number of factors, such as feedback related to the product or vendor, popularity, listing interaction from the clients, as well as the buyer’s browsing history on the site. For example, in the case of the *Cartel Marketplace*, the implementation of this mechanism is called *Cartel PageRank*, it is awarded to the product, and the higher it is, the more traction a product will get. The *White House Market* also has a similar mechanism in place, which moves the top 20 sellers, based on the amount of sales in the last 45 days by Monero (XMR) value, higher up the product list. Lastly, vendors can choose to pay for the promotion of their products (see Section III-F4) by issuing a fee to the marketplace, instead of letting the algorithm do it for them, by factoring in the aforementioned variables. With the *White House Market* again as an example, vendors can bid for eight spots, rotating every single week, where their products can be featured.

2) *Currency:* The most popular and most widely used cryptocurrencies to conduct payments in darkweb marketplaces, are BTC and XMR. The differences between the operation of the two protocols, have great impact on the level of anonymity that they are able to offer to their users.

a) *Bitcoin (BTC):* The main issue with the usage of BTC has been privacy. Transactions made with BTC can be monitored, due to the fact that they are publicly announced on the blockchain. By using a block explorer, one can easily find information about payments made to certain wallet public addresses, along with their origin, the exact amount transferred, transaction history and balance. This leads to Bitcoin having a *fungibility* issue [16], meaning that two BTC coins can never be regarded as equal, since every BTC can be traced back to its point of creation in a defining way. Furthermore, acquiring BTC from a cryptocurrency *exchange*, will require providing some kind of identification, also known as *Know Your Customer (KYC)* information. The combination of these two facts, can potentially lead to the deanonymization of users, in the event of a marketplace seizure. In such a scenario, gaining access to the marketplace’s wallet, could lead to LEAs following the trail back to the public address (or addresses in the case more than one are being used) of a buyer, which can then be linked to the user’s real identity, through the information available to the exchange service. In an effort to make BTC more anonymous, *mixers*, or also known as *tumblers* [17], [18], came into play, which aim at erasing the trail the transactions leave behind, for a small fee. One simple example scenario, would be making a payment to the mixer service, which would “mix” the funds with those of other users, and then transfer the amount to the desired destination wallet address. With the mixer acting as the middle man, the trail that could lead back to the original user, is harder to follow. This mechanism can also be used to launder BTC, where a user could send the funds to the mixer, and then have the mixer transfer the funds back to them, after the “mixing” process is complete. However, similarly to how exchanges operate, mixer services will often keep information about their users, which can be used to trace back to the user a transaction originates from. Additionally, this way of operation is very

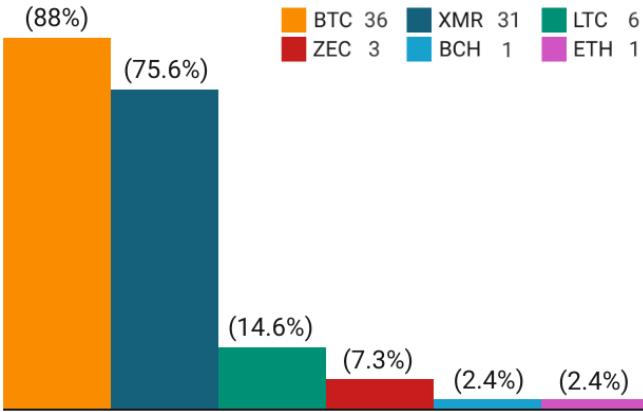


Fig. 2. Cryptocurrency adoption on the 41 darkweb marketplaces. The legend depicts the total number of marketplaces out of the 41, that allow for transactions with each cryptocurrency.

prone to phishing attacks and scams, which translates into fake service providers, that will keep the funds for themselves after the user has transferred them.

*b) Monero (XMR):* Despite Bitcoin's popularity over the years, the most recommended and safest practice to carry out payments on the darkweb, is through the usage of Monero. XMR obfuscates the origin, destination, and amount of the transactions, which makes tracking them back to users extremely challenging for LEAs [19], [20]. XMR also has the *fungibility* property [16], contrary to BTC (see Section III-B2a), making coins interchangeable. Marketplace users are encouraged to swap out any BTC they own for XMR and then move forward with their purchases. Even in the event that a user would like to make a transaction specifically using BTC, the recommended course of action, deducted from guides and discussions on the darkweb, is to initially convert BTC into XMR, then convert again to BTC using a second independent wallet, and only then go through with the transaction. Both of these practices are effective, due to the fact that the trail is lost the moment the BTC amount is converted into XMR. Despite the additional security that XMR offers, marketplaces seem to be making an effort to make transactions even more safe for users, with *AlphaBay* as an example, which uses XMR mixers as an extra layer of security.

Bitcoin and Monero might be the most frequently used cryptocurrencies at the moment (July 2021), but there are certainly others being used as well, namely Litecoin (LTC), Zcash (ZEC), Bitcoin Cash (BCH) and Ethereum (ETH). In Figure 2 we present the percentage of cryptocurrency usage throughout the 41 marketplaces we investigated.

Lastly, in the particular case of *Hydra*, there are a few additional methods of payment. The Russian marketplace also accepts payments through the *QIWI Wallet*, which allows for anonymous transactions, and through SIM card top-ups [21].

*3) Wallets:* Payments on the marketplaces, can be done either through off-site wallets, or through each platform's on-site wallet (if it utilizes one). Regarding off-site wallets, darkweb guides and forums threads advise users against using

*custodial/hot wallets* to store their cryptocurrency. In this scenario, the user shares custody of their private key with a third party, and since the user does not have exclusive control over their private key, the same can be stated about their funds ("Not your keys, not your coins." [22]). Conversely, *non-custodial/local wallet* usage is encouraged, such as hardware wallets, where the private keys are saved locally and are never shared with anyone [23]. In the case of a purchase, the user deposits the amount to a cryptocurrency address, unique for every purchase.

With on-site wallets, users can deposit funds in various cryptocurrencies, depending on the platform, and credit them on their accounts. This is done through a cryptocurrency address, generated by the marketplace, which is unique for every user, and usually available for a certain period of time (e.g. 7 days). Users can then use these funds, or account *balance*, to carry out their purchases, instead of using their own wallets. The users can still maintain their own wallet as mentioned above, from which they perform the balance top-ups. Each user has the option to withdraw their balance at any point, but the process differs per marketplace (see Sections III-F3 and III-B6). Purchasing via an on-site wallet, translates into the funds being redacted from the account balance. Some platforms, e.g. *Darkode*, have an on-site wallet, but also allow for payments through the users' wallets.

In the context of wallets, one mechanism that stands out among the marketplaces, is *AlphaBay's AlphaGuard*. In the case that the marketplace is seized by law enforcement, this mechanism will broadcast a set of onion and Invisible Internet Project (I2P) addresses through various channels on the web that the users can visit and retrieve their funds (these channels are not however specified). This is done through a key that each user is given by the marketplace at the point of registration called the *wallet recovery key*, and along with their username and password, it can be used to empty the on-site wallet by moving the funds to a new deposit address chosen by the user. The description of this mechanism is in line with the information provided by the marketplace owner themselves. We were not able to test it, since it would require an attack against the marketplace to take place, but we were however provided with a wallet recovery key after the registration process.

*4) Escrow:* *Escrow* is the primary mechanism through which darkweb marketplace sales are carried out. When a client purchases a product, both in the cases of an on-site and an off-site wallet, the paid amount is transferred to a cryptocurrency wallet owned by the marketplace. The wallet public address used to deposit the funds, is unique for every purchase. The amount will remain there up until the client verifies that they have received the product they purchased. Only then will the amount be released from the wallet and transferred to the wallet of the vendor. This system aims to avoid incidents of fraudulent behavior from the side of the vendors. In the case of its absence, the marketplace would not be serving as an intermediary, meaning that the vendor would be directly paid by the customer. In this scenario, it would be

very easy for the vendor to lie about dispatching the ordered items, or even simply cutting all ties with the client altogether, while having received the paid amount.

5) *Auto-Finalize*: From the vendor's side, in order to offer some type of assurance that the funds will eventually reach them, even if the client does not notify the marketplace about the delivery of the order, marketplaces implement the *Auto-Finalize* mechanism. This mechanism dictates that after a set time interval (e.g. 14 days, or even 45 days), if the client has not verified the delivery of the ordered product, the order will be finalized automatically, releasing the funds from escrow to the corresponding vendor. Failing to finalize the order, in some marketplaces, will lead to the client account receiving negative rating, incentivizing users to finalize as soon as possible.

6) *Multisignature Escrow (Multisig)*: One issue that still remains, despite the implementation of the escrow mechanism, is *Exit Scams*. In an escrow purchase scenario, since the paid funds are initially transferred from the client to the marketplace wallet, the whole amount is under the control of the marketplace. Since the rise of darkweb marketplaces, there have been many incidents, where the marketplace would shut down, going offline without prior notice to the vendors and clients. All the funds gathered from every single purchase carried out, would remain with the marketplace owners, with vendors left unpaid, and some of the clients paying for a product that will be never dispatched (some orders might have already been on the way to the clients). Some examples of exit scams are those of the *Wall Street Market*, *Icarus Market*, *Elite Market* and *Empire Market*, with the last one, shutting down and stealing \$30 million worth of bitcoin in the process [24].

To eliminate the danger of exit scams, many marketplaces support BTC multisignature escrow payments, or *multisig* (see Appendix C). In a multisig scenario, the main idea is that out of the three entities involved in a purchase, namely the buyer, the vendor, and the marketplace itself, there is always authorization from two of them needed, to finalize a transaction. This effectively means that in order for the transaction to be completed, the corresponding private keys will be needed, to provide the two necessary signatures. Typically, one signature will come from the client, after they have received the product, and one from the vendor themselves. In the case that the client does not finalize the order after receiving the item, the marketplace and vendor can provide the two signatures. Most importantly, in the event of a marketplace exit scam, the funds are not trapped inside the marketplace wallet, which is the case with the standard escrow paying scheme. The funds can be released through common understanding between the vendor and client, agreeing to both sign off on the transaction, and let the purchase process reach finalization without further issues. The same practice applies when paying for marketplace commission fees (see Section III-F1).

7) *Direct Payments*: Direct payments were not implemented on either of the marketplaces we explored, contrary to vendor shops, for which this was the only available payment option. One exception to this rule, was the *Televend* marketplace. Televend uses the *Telegram* application, as a

platform to carry out sales. Users can join a channel, and purchase their product of choice, directly from the vendors. The hidden service site is only utilized to present information such as reviews, feedback, vendor profiles, listings, and to provide vendor registration and verification, making the Telegram channels the actual marketplace. The purchase process is automated through the deployment of Telegram bots, and without utilizing any type of escrow mechanism.

8) *Finalize Early (FE)*: Many marketplaces, have a mechanism in place, which allows for the transfer of the paid funds to the vendor, immediately after the payment has come through from the buyer, before the ordered items are even dispatched. This mechanism is called *Finalize Early (FE)*. Early finalization aims to provide ease from the vendor's side, who does not have to wait for the order to arrive to the client to receive the payment. Additionally, in case a client fails to finalize the order, the vendor does not have to sit through the whole duration of the escrow wait time, which can sometimes be more than a month. Marketplaces will only provide the finalize early badge/capability to highly trusted vendors, replacing standard escrow. For example, *World Market*, which is a very popular marketplace, will only assign the label to vendors that have reached the "level 5", which requires 250 sales, \$25 000 in sale volume, and 90% positive feedback from past clients. Some marketplaces will also take into consideration sales, reviews and feedback from other marketplaces that the vendor has been making sales on, as well as whether they already have achieved the FE verification on other platforms.

The question that naturally arises from the implementation of the FE functionality, is why should a client want to buy from such a vendor, since due to the absence of escrow, they essentially have no fall-back in the case their order never arrives. There is no assurance that the product will even be dispatched in the first place. The answer comes from the requirements that a vendor needs to fulfill to achieve the FE status. Having this status is on itself a guarantee that the vendor is well established, verified, offering high quality products, and held in very high esteem in the darkweb marketplace ecosystem. The probability of scams from these vendors are extremely low, since no vendor would risk damaging their reputation, that they worked so hard to build. Additionally, in many occasions, in order to motivate the buyers and make buying from FE vendors more appealing, marketplaces will offer some kind of discount. An example is the *Cartel Market*, which applies a 5% discount to orders from such vendors. This serves vendors, marketplaces, and buyers alike. Vendors, make more sales, which directly means more profit. Marketplaces are hosting these sales, which translates into more commission fees from each sale (see Section III-F1). Lastly, buyers get a better price for the product of choice, which will be of higher quality, because of the prestige that accompanies the FE status.

9) *Refunds*: In the case of a transaction running into issues and a refund is necessary, the buyer will provide a cryptocurrency address, where the funds will be deposited, or in the case of an on-site wallet, the amount will be credited to their account balance.

### C. Shipping & Delivery

The details surrounding the dispatch of a physical product, and its delivery to the buyer, are a determining factor in how cost efficient and discrete a purchase from the darkweb can be. Vendors will list the shipping methods available and the client is free to choose whichever they prefer, but there are many details that can make the difference between a successful delivery and prosecution by the law. It should also be mentioned, that regardless of which of the following methods the client chooses to use, any private information given to the vendors, such as names and addresses, are always encrypted through the PGP protocol.

1) *Origin & Destination Countries:* The first determining factor regarding the risk taken when purchasing from a darkweb marketplace, is the country of origin, that the product will be shipped from, as well as the country it will be delivered to. Ordering from foreign countries, carries far greater risk than doing it domestically. The main reason behind this, is the fact that the product will go through customs twice, once leaving the country of origin and once entering the destination country, increasing the probability of the order getting intercepted. Many users are tempted to place an order from outside their countries, due to the fact that in the majority of cases one can find the same product at a lower price from non-domestic vendors. Furthermore, according to past experiences from marketplace buyers found on forums and guides, packages arriving from certain countries are labeled as more probable to contain illegal items, with some examples being the Netherlands and Colombia, in connection to drug trafficking. Ordering from these countries will certainly carry greater risk for a buyer, since the package carries more suspicion than usual. Similarly, some countries have more strict custom checks, with Sweden and Norway as examples, making packages ordered internationally, while being a citizen of these countries, more prone to getting intercepted at customs.

2) *Real Name & Address:* Throughout forums and marketplaces, discussions and guides point to the same practice, when it comes to placing an order. The users are always encouraged to use their real private information, namely their names and addresses. The main reason behind this course of action, is that not doing so, is considered much more suspicious behavior. Handling an order from a darkweb marketplace, the same way one would treat an order from a legitimate online shop, is much less likely to draw any attention. It is considered that even if something goes wrong with a delivery and a package is intercepted by LEAs, as long as it cannot be proven that the order and payment were carried out by the buyer, then the buyer is safe from prosecution. This applies even in the case that the buyer signs for the delivery, making Operational Security (OPSEC) of the utmost importance in both scenarios.

3) *Post Office (PO) Boxes:* Another available option for buyers, is using a Post Office (PO) box. Creating a PO box requires a real name and address, tying the user's identity to its existence. Using a fake ID is strongly advised against, since it is much more likely to create suspicion. By registering a PO box the buyer loses plausible deniability, since the box is

registered under their name, and unlike their address which is public, it is private. Hence, the majority of forum user posts, pointed towards avoiding the usage of PO boxes (see Appendix A), and many vendors will not list PO boxes as a delivery option, considering this method to be an OPSEC risk.

There were two more interesting practices mentioned on the forums. One was using fake IDs to open PO boxes in "mom-and-pop" shops (small family business shops), which the users should close after receiving their order. They would then repeat the same process on another shop, with a new box. The second method was UPS store boxes. In this case the buyer registers for PO box, but under the pretense that it is to be used for an online business, sidestepping the requirement to provide their real name. Instead, they provide a fake business name, which cannot be tied directly to the user.

4) *Drops:* Apart from having the package delivered to their house, a user can also choose to use a *drop*. Drops are in essence locations that cannot be related to the buyer, but can still be used to receive mail. Guides on the darkweb explain how to choose the optimal location, as well as how to make it look as less suspicious as possible. An example given, is choosing an uninhabited house, at which the user should go from time to time, without making themselves memorable, but creating the belief to the rest of the neighborhood that there is someone associated with the premises. A guide even mentioned performing some kind of maintenance on the grounds, such as mowing the lawn. Nonetheless, the main suggestion was that the user should send mail to that address using an alias, as a means of "priming" the address. This would help towards not drawing any unwanted attention when the marketplace order finally arrived in the mail. However, drops are generally discouraged, since as previously mentioned, using the real address and name is the safest option.

5) *Dead Drops:* Some vendors will also provide *dead drops* [25], [26] as a means of delivery, which was initially documented on the *Hydra* marketplace in 2014. In this scenario, the purchased item is left at a random location, that only the buyer and the vendor are aware of. No names or addresses are exchanged, maintaining anonymity for both parties, and sidestepping the dangers associated with normal post. These locations can be anything from remote spots, like a specific tree in a random street, to very public places, such as public transport stations. The item drops are handled by individuals known as *droppers*, who get paid on commission depending on the type and amount of the product they deliver [26].

The execution of a dead drop can be summarized into a few simple steps: finding the perfect location, placing the item, taking a picture on which the exact spot where the item was placed is marked, and lastly, including the GPS coordinates along with a map screenshot of the exact location. After the drop is made, the dropper will upload all the information on the marketplace, so that the buyer can use them to retrieve the package [26], [27].

6) *Packaging:* Another determining factor on whether a delivery will be successful or not, is packaging. Packaging can easily be the cause of a delivery drawing unwanted attention,

and getting intercepted by LEAs. For this reason, discussions on forums, along with previous research on the subject [28], point to certain practices, that are utilized to avoid detection, through eliminating smell and DNA traces, that could be left on the package. These practices are air-vacuuming the item at least once, use of heat-sealed bags/Moisture Barrier Bags (MBBs) and Mylar paper, printed labels, use of decoys for external packaging, in which the item can be hidden, and cleaning the packaging with alcohol. Furthermore, data points towards the use of specific gear while packaging the items, such as cotton and latex/rubber gloves, used in combination with one another, long sleeve shirts, hairnets, ski masks, even full body protective suits, such as hazmat suits. One more practice suggested, is using a different room to externally package the item, than the room in which the product is held, which in the case of drugs, could potentially contaminate the packaging, making it prone to detection. Information on the darkweb suggests making a compact list, of all of the above methods that a vendor could use to package an item before delivery. This way, the vendor would be less likely to make a mistake, making the whole process of shipping safer.

In the case of firearms, vendors have been documented to ship the weapons disassembled, in different packs and through different postal services, including an assembly guide [29]. Additionally, in order to conceal the products, most vendors will use unorthodox methods of packaging:

*[Purchased products are concealed] ... "In Computer devices; In cans never opened; In air freshener or coca cans; In books; In stoles of pairs of shoes; It may come in bottles; In all kind of Computer devices; In Electrical goods; And in all kind of products." - [30]*

One can also find 3D printing plans for firearms and their parts, listed as digital products [29] (see Section III-C9).

7) *Return Address*: Not including a return address on the package, or using fake addresses or names, can cause suspicion and draw unwanted attention to the package. For this reason, it is often recommended that vendors use either a real address and name belonging to random individuals, or a business, preferably small. In the first scenario, vendors are even encouraged to use the information of people living in neighborhoods with a bad reputation. The justification for this is that in the case of a returned package, it is supposedly less likely that the package will be reported to the police. Vendors also have the option of using the return addresses belonging to businesses or shopping centers, but in combination with a fake minor identifier, such as office or floor number.

8) *Tracking*: Users are also advised against tracking their order, unless it is provided freely by the post service, since in this scenario LEAs cannot prove that the order is actually related to the user. In both cases however, buyers are strongly discouraged to use this feature, since it can leave traces.

9) *Digital Products, Autoshops & Automated Vending Carts (AVCs)*: In the case of digital products bought on the marketplaces, the process becomes much simpler. The methods of shipping include sending a message to the buyer, by

using the built in platform messaging system encrypted with PGP, attaching a file containing the product, or providing a download link. Digital items can also be sent via e-mail, and in the case of debit card, or PayPal account balance, they can also be delivered directly as a transfer to a bank account, PayPal account, or cryptocurrency deposit. Users can also use cryptocurrency to acquire transfers through Western Union.

Some marketplaces will also implement *Autoshops*, which aims to make digital purchases faster, by eliminating the escrow mechanism. The funds are directly transferred at the moment of purchase, following the *finalize early* mechanism (see Section III-B8), thus making the purchase process instantaneous. After the payment is complete, the buyer receives the digital product through the same channels mentioned above.

Lastly, it should be mentioned that there is a special type of platform offering digital products, known as AVCs [31], which function entirely automatically, and one could in essence describe them as standalone autoshops.

#### D. Vendor Reputation

A vital element regulating the entire darkweb marketplace ecosystem, is trust. Vendors' reputation, has a great impact on their financial success, since it is the primary contributing factor towards building the trust of potential buyers. The darkweb can often seem a scary place, with users feeling hesitant to go forward with purchases, or even visit certain platforms. Being scammed by darkweb marketplace vendors is quite common, when their reputation is not taken into account by buyers. By creating a safe environment, users are encouraged to trust the vendors and carry out purchases. Since trust appears to have such a great influence on every individual associated with these platforms, marketplaces have implemented certain mechanisms that aim to build that trust, and make sure it does not get compromised at any point in the future.

1) *Reviews & Feedback*: Similarly to legitimate online platforms, reviews and client feedback also play a leading role in shaping the reputation of a vendor. Users who have purchased from a vendor, are given the option to post a review based on their experience. This review can be on the vendor themselves, or the specific product. Furthermore, buyers are given a specific time window after the purchase (e.g. 14 days), in which they can submit their evaluation. After this time window elapses, the evaluation cannot be changed.

Due to the importance of reviews in shaping the reputation of vendors, some marketplaces have systems in place, which aim to eliminate fake review instances on their platforms, such as the *Fake Review Detector* of the ASAP marketplace. Lastly, evaluations can also be found on darkweb forums, contributing to shaping the opinion around a vendor through reviews and discussions between past buyers (see Section III-E).

2) *Reputation Classes & Cross-Platform Reputation*: *Classes* are one of the main mechanisms used on marketplaces to inspire trust to users. The implementation of these mechanisms, varies per marketplace but the notion remains the same: the higher the verification level of the vendor or product, the

more confidence it instills to potential buyers. Furthermore, vendors can be often individually evaluated on individual elements such as overall quality of their products, shipping, responsiveness, communication and labeled as a source of “value for money” products, all of which establish the level of trust, efficiency and ease, that comes when associating with that vendor.

The most common applications of this system, is vendor *levels/ranks* (e.g. from 1 to 3, 1 to 6, or 1 to 10), which is derived from the number of sales carried out on the marketplace. A *star* system is very similar to the “1-5” system used in clearweb online shops, which is most commonly calculated from the user reviews of the vendor, based on their experience. It can also be applied to products, based on their individual client reviews. *Statuses* are used by the *Televend* market, which provides a very detailed overview of the requirements necessary for each status to be appointed, namely vendor time of operation, positive reviews, and sales. They start with the new vendor status, then verified, established, trusted, elite, veteran, and lastly legendary status. Some other mechanisms used are *tiers* (e.g. bronze, gold, diamond), color based ranking, positive feedback percentages, and the *Finalize Early* status (see Section III-B8). One more metric that can be used, is the ratio between disputes won and disputes lost (see Section III-E3), as well as the amount of total disputes filed against them by buyers, which will be included on their profile along with their class. Having a poor ratio, or a large number of disputes, impairs the vendor’s chance at reaching high reputation and finally receiving the FE badge. It also rises suspicion from the side of potential clients, regarding the vendor’s practices. Furthermore, there are cases that vendors are ranked separately on different aspects of their business (see Section III-D1), and in combination with the received client feedback and dispute resolution statistics, they are assigned an average ranking, which can be in any of the forms mentioned above. Reviews can also serve as a graphical representation of clients’ feedback on a specific product, with the *Cartel Marketplace* as an example, which uses a bar filled with green, yellow, and red blocks, underneath the product, to illustrate the positive, average, and negative reviews, respectively.

Some marketplaces allow for the activity of the vendor in other marketplaces to be included in the calculation of their ranking, after the vendor proves their identity. Vendors can maintain the same username across platforms, if they provide the proof required (e.g. PGP key), which aims to help them preserve the reputation that has already been built around that username, their clientele, as well as make them more attractive to new buyers. This translates into vendor information that can serve as criteria to assign them a ranking, being available across different platforms. This led to another ranking mechanism, the marketplace verification levels. These levels are assigned according to how many marketplaces can vouch for the specific vendor. For example, if a vendor is already high ranked in three marketplaces, a new fourth marketplace can assign them the verification level “3” when they join the platform, and they will be awarded the “verified”

badge, if their status reaches a high verification level. The contribution of vendor information taken from marketplaces that were seized by LEAs at some point in time, in most cases, was found to persist and still be counted towards the vendors’ verification levels after the marketplaces’ takedown. It should be mentioned that the exact formula that is used to assign trust levels to vendors, namely which specific vendor characteristic’s and performance statistics are taken into account, as well as their individual impact, are often left undisclosed by marketplaces for security reasons (e.g. *AlphaBay Marketplace*).

Lastly, it should also be mentioned that gaining a high verification level on a marketplace, also affects the position of a vendor’s listing in the search results, in the corresponding product category. This raises the probability of users purchasing the product, which drives the vendor’s sales up, contributing towards their verification level going even higher, and placing their listings high on that product category yet again, creating a cycle.

3) *Harm Reduction*: With the *White House Market* and the *DarkOde Reborn Market* as the first to implement it, the *Harm Reduction* initiative aims to mitigate the dangers that can occur from drug impurity. According to this mechanism, vendors can include a testing kit in their listings, which then the buyer can use to test the product they received and evaluate its quality. They can then submit the test results on the marketplace through a dedicated form, along with a review of the tested product signed with their PGP key, and a photo showing the product, the results, the vendor name and the date. They can also post this information on the *Dread* forums sections */d/HarmReduction* and */d/Reviews*, as well as on the sections dedicated to each marketplace, namely */d/WhiteHouseMarket* and */d/DarkOdeReborn*. Posting a test result, will earn the reviewer a *Quality Tester Badge*, and doing so regularly, will lead to earning perks, such as gift cards that can be used for purchases on the marketplace.

From the side of the sellers, vendors who receive 1 positive test result for their product, will earn the *Product Tested* badge on the product page. Receiving 3 positive test results, will lead to the vendor being awarded the *Quality Vendor* badge, which is shown on the vendor profile, while earning positive test results systematically will give the opportunity to the vendor to apply for a reduced commission fee from the marketplace. Lastly, harm reduction listings will gain priority over normal ones in the *Featured* listing feed of the marketplace.

The harm reduction mechanism, apart from more safety for the buyers, also leads to more profit, for both the vendors and the marketplaces. Vendors offering high quality products on these marketplaces will lead to more and more people trusting them and their products. This trust will then lead to customers being more encouraged to choose these vendors over others, leading to more purchases on the marketplaces that host them. Consequently, the marketplaces’ profits will also increase, since more purchases carried out through the platform translates into more commission fees.

## E. Support, Disputes & Community

Mechanisms that are meant to handle any arising issues, regarding purchases from the marketplaces, have great impact on how successful, profitable and popular a marketplace will eventually become. Knowing that there will be assistance from the platform when needed, creates the feeling of safety to the buyers, making them more inclined to use it, constantly growing the marketplace's client base.

1) *Support Staff:* The majority of marketplaces, make sure to have dedicated support staff in place, that will assist users deal with any challenges they might face. This is quite often stated clearly on each platform, or even advertised, since it plays such a important role in its smooth operation. Users are able to create support tickets, explaining the challenge they are facing, which will be addressed by the support staff. In some cases there will be an automated support bot, which will initially try to resolve the situation, and if that fails, the user will be redirected to a staff member. Furthermore, the staff is usually composed of individuals speaking different languages, and are available in a variety of time zones, in an effort to accommodate for the different geolocations that the clients might be located in, and provide 24/7 assistance.

2) *FAQ Sections:* Since darkweb marketplaces implement so many mechanisms and are composed of so many different elements which regulate their operation, they deploy FAQ sections which aim to assist users use the platform, as well as inform them of the rules they need to follow. Depending on the marketplace, FAQs can provide information regarding the rules regulating purchases, selling, payments, and any other basic piece of information needed from buyers and vendors to use the platform, including guides on some of the implemented mechanisms, such as PGP and 2FA.

3) *Disputes:* In the case of a buyer not being satisfied with the way their purchase was handled by a vendor, they can create a dispute. In a situation like this, the buyer will create a ticket explaining what has gone wrong with their order from a vendor, and the support staff of the marketplace will try to handle it<sup>4</sup>. A dispute will usually be created due to issues related to shipping, such as longer delivery times than expected/no delivery, in combination with the vendor being unresponsive. It can also be related to the state of the delivered product, such as receiving a different product than advertised, a damaged product, or a lesser amount of the product than paid for, and it can only be created for a specific time period, which in most cases is a few days before the auto-finalize is executed. In general, most marketplaces propose users should initially try to solve all issues they might run into, by contacting the vendor directly. If that fails (e.g. unresponsive vendor), they are then encouraged to submit a ticket, creating the dispute, and getting the support staff involved to resolve the situation. This process does not apply to purchases from FE vendors, since the order is considered completed the moment the payment is completed, meaning that the buyer forfeits the

<sup>4</sup>The *AlphaBay* marketplace, has successfully managed to automate the dispute solving procedure, by creating the *Automatic Dispute Resolver (ADR)*.

right to dispute. Marketplaces strive towards their users not creating disputes lightly, so they explicitly warn them that if they end up losing a dispute, they will receive negative feedback/rating from the marketplace administrators.

As mentioned in section III-D2, dispute resolutions can greatly impact the reputation of a vendor. The lost/won ratio of a vendor's disputes, as well as the number of total disputes filed against them, are all taken into consideration by marketplaces in the process of rank appointment, making the dispute mechanism very effective in the marketplace's effort to keep vendors' operation in check.

4) *Forums:* Forums' importance in the darkweb is vital. They are a place of discussion on various topics, with one of them being marketplaces. Potential buyers can easily browse through these discussions between former buyers that evaluate, promote, criticize vendors, and report scammers, helping newcomers to assess the risks when choosing to buy from a vendor. In addition, they include guides on some of the more technical aspects of using the darkweb, such as PGP encryption, cryptocurrency payments and 2FA. Forums are also used by vendors to advertise their services, by clients looking for a specific product/service, as well as by marketplaces promoting their platform, and making various public announcements. Specifically, marketplaces can choose to have an individual integrated forum, or use *Dread*, with a section dedicated to their platform. An example is *DarkFox Market*, which uses a dedicated section of *Dread*, called */d/DarkFoxMarket*, for posts related to the marketplace.

5) *Communication:* Communication between vendors and buyers, is to be carried out through the platform itself, for which marketplaces will mostly utilize the PGP protocol. Vendors are specifically forbidden from listing any other means of contact in their product listings or profiles, such as *Jabber/Extensible Messaging and Presence Protocol (XMPP)* or *Wickr*, in combination with the marketplace's policy of not conducting sales off-marketplace. There are cases that the marketplace itself will provide an alternative communication mechanism, which is very often a *Jabber/XMPP* server in combination with *OMEMO Multi-End Message and Object Encryption (OMEMO)*, *PGP* or *Off-The-Record (OTR)* encryption, dedicated to fulfilling the platform's needs.

Despite the fact that the properties of vendor shops are a subset of the ones found on marketplaces, the means of communication used differed between the two types of platforms. In more detail, vendor shop owners, apart from on-site contact forms, were found to also include messaging applications such as *Telegram* and *Wickr*, or preferred communication via encrypted email services such as *Mail2Tor* or *ProtonMail*.

## F. Marketplace Revenue

In this section we document four main sources of income for marketplaces: *purchase commissions*, *the vendor status*, *withdrawals*, and *listing promotions*. In addition to these sources, some marketplaces will also deploy certain mechanisms which aim to keep the users engaged and motivated to keep using

the platform, while in some cases also receiving commissions from their usage by clients (see Appendix B).

1) *Purchase Commissions*: The role that marketplaces play in the darkweb trading ecosystem, is serving as a platform where vendors can list their products, and buyers can easily browse through and carry out purchases. The owners of these platforms do not actually sell any products, so their profits and economic incentives to run a marketplace are not sales. A basic source of income is commissions. Marketplaces will require a fee from vendors (in some cases from buyers as well), which is a percentage of the total amount paid for the purchase. In most cases, commissions range from 3% to 6%, but with some caveat. Varying per platform, commissions are either a standard fixed amount, or fluctuate depending on the price paid, the amount of product purchased, whether the purchase was carried out using the escrow or multisignature escrow mechanism (*DarkFox charges a 5% fee for normal escrow and 4% for multisignature escrow*), as well as depending on the rank of the vendor (e.g. lower vendor rank, translates into a higher commission paid to the marketplace).

2) *Vendor Status*: Another source of income for darkweb marketplaces, is granting the vendor status. Individuals interested in becoming vendors, have the option to upgrade their accounts by paying a fee, the vendor bond. This fee varies per marketplace, and it mainly lies between \$100 and \$500, but can exceed that depending on the level of prestige and reputation of each marketplace, even reaching the \$1500 margin in the case of the *World Market*. Depending on the marketplace, the vendor bond can also be refundable. In addition, some platforms also require proof of product in order to provide the status, or even that the total value of the available products, amount to or surpass a specific price margin (e.g. *Hydra* marketplace will only grant the status if the cost of all goods is over \$400). However, some marketplaces do not require a fee to provide the status, as long as the individual in question can provide proof of past experience as a vendor on other platforms. Lastly, *Hydra* does not follow the vendor bond scheme, and instead of a fixed fee, it requires a monthly subscription or “rent” from the vendors. The price for this rent begins at \$400 per month, but can drop down to \$125, if the vendor chooses to opt for a 12-month prepayment.

3) *Withdrawals*: Some marketplaces also require withdrawal fees, which are applied every time a user wants to make a withdrawal from their on-site wallet balance (see Section III-B3). This fee can be a fixed amount, like in the case of *World Market*, which applies a flat 0.0003 BTC rate ( $\approx \$14$  in August 2021), or a percentage of the amount withdrawn, with the *DarkOde Reborn* marketplace as an example, which applies a fixed 2.5% rate. In some cases it can also be a combination, where a flat rate would apply up until a specific amount, and then a percentage rate is applied from that point onward. An example is the *Liberty* marketplace, which applies a flat \$1 rate for purchases up to \$100, and then a percentage rate of 1%, for every transaction over that \$100 margin. Some marketplaces will also have a limit set, regarding the minimum amount that users can withdraw, and the minimum they can deposit. An

example is the *DarkFox Market* which has a 0.00005 BTC ( $\approx \$2.3$  in August 2021) minimum limit for deposits, and a 0.0005 BTC ( $\approx \$23$  in August 2021) minimum limit for withdrawals.

4) *Listing Promotion*: An additional source of income for these platforms, is the fee paid by vendors to promote their products, as discussed in section III-B1. Most marketplaces will assign a specific number of listing spots, which will be positioned higher than any other listing, on the homepage of the marketplace. These listings are usually called *Featured Listings*, and are more likely to get higher traction by clients, since these products are the first that a visiting user will see. Vendors can bid for these slots, and if they win the auction, they are then able to use the listing slot for a certain time period. In the case of the *AlphaBay* marketplace, this time period is two weeks, and the slots available every week are eight. The auction for the next listing slots also lasts two weeks, until the expiration of the previously auctioned slots. In the case of *White House Market*, the winning bid for a featured listing slot has been known to range from \$2000, up to \$3000 per month.

A similar mechanism implemented by marketplaces to promote certain listings is *Sticky Listings*. In this case, vendors can pay a fixed fee, in order for their product to get priority over others in the search results of certain product category, for a certain time period. One example is the *White House Market* which charges \$300 per week, for each sticky listing. In addition, some marketplaces will provide free sticky listings to some new vendors randomly, to help them kick start their business.

#### IV. RELATED WORK

Darkweb marketplaces, have been targeted by researchers with various approaches, all aiming at gaining a deeper understanding of the darkweb trading ecosystem.

*Nunes et al.* [14], with the purpose of acquiring cyber threat intelligence, developed a system which would harvest information from the deepweb and darkweb. This system consisted of a crawler, a parser and a classifier, and it was used to gather data from 17 marketplaces, as well as 21 forums. They also illustrate two case studies, one on the discovery of zero-day exploits sales on the marketplaces, and one on the presence of vendors in both forums and marketplaces, using the data acquired from both types of platforms. *Nicolas Christin* [32], carried out a measurement analysis on the Silk Road marketplace, over a period of 8 months in 2011-2012, before its shutdown took place. Using daily crawls, an effort which spanned 6 months in 2012, he gained insight on the marketplace’s operation, presenting data on elements of the marketplace such as products, sales, vendors, and customer feedback. Additionally, he discusses the role and importance of BTC, in the marketplace’s operation. Building upon this work, *Soska and Christin* [13], study the growth of underground marketplaces from 2013, when the Silk Road marketplace was taken down, until 2015. They collected data from 16 marketplaces, which contributed towards understanding how the underground marketplace ecosystem operates, from the

types of products available and their evolution, to vendor presence throughout the darkweb, as well as security mechanism deployment, such as PGP.

All three of these efforts, try to unveil the darkweb marketplace infrastructure, but take a more quantitative approach compared to our work, with crawling and its resulting dataset, being the main point of focus. *Thomas S. Hyslip* [12], takes an approach more similar to ours, and illustrates the framework that surrounds the trading of digital services and products on marketplaces, while we explore the broader spectrum of products and features. *Kermitsis et al.* [33], also touch upon the characteristics and properties of darkweb markets. Conversely, our work is mainly founded on real-life applied information, such as advice and guides from popular vendors, as well as user experiences narrated on forums. We also focus more on in-depth insight on the individual properties and practices of these platforms, as well as the reputation element, which is arguably a vital factor of this framework's successful operation, creating trust between vendors, buyers and marketplaces. Lastly, there has also been research targeting specific product types, such as drugs [28], [34], [35], firearms [29], [30], [36], as well as COVID-19 vaccines and proofs of vaccination [2], [15], contributing towards investigating the different characteristics associated with each type of illegal trading.

Apart from the darkweb illegal trading framework, clearweb marketplaces and forums have also been targeted by researchers, with the same goals in mind. Despite the fact that these platforms operate in the clearweb, the methods implemented also apply to darkweb platforms due to the similarities of the two markets (e.g. trust, reputation, anonymity).

*Pastrana et al.* [37], developed the *CrimeBot* crawler, which was utilized to scrape underground forums, in an effort to better understand the behavior of individuals involved in cybercrime, as well as the ways that potential cyber criminals are incentivized to enter the cybercrime world. The data was harvested in a period of over 9 months, and was used to create the *CrimeBB* database. This database includes more than 48m posts, from 1m accounts of 4 forums (2018), with some posts dating back to 2005. Lastly, they present a case study on the evolution of currency exchanges, to illustrate the dataset's potential. *Hutchings and Holt* [38], investigate the infrastructure of the stolen data markets, through crime script analysis. Using qualitative methods, they examine the content of 1,889 communication instances between sellers and buyers, from 13 forums that operate as selling points for stolen data. *Holt and Lampke* [39], also work in the same direction, employing qualitative procedures to analyze 300 threads from six forums dealing in stolen data. *Holt et al.* also focuses on the element of trust, in the context of stolen data markets. The importance of the role that trust and reputation play in the illegal trading world, both in the darkweb and clearweb, is crucial, making research towards this topic of great value. Last but not least, *Vu et al.* [40] use the *CrimeBB* [37] dataset, containing 190,000 user contracts, created from June 2018 to June 2020, from one of the most popular forums *Hack Forums*,

to perform an longitudinal analysis of the platform's operation. They illustrate how the forum's operation has evolved over this two-year span, from an economic, social and reputation/trust standpoint, split into the three distinct time periods, namely the period the contract was adopted (set-up era), the stable operation era, and finally the COVID-19 era.

## V. CONCLUSION

Illegal trading on the darkweb owes its success to a combination of properties. Marketplaces deploy mechanisms that aim to provide ease of use, security, obfuscation, resilience against hostile actions, along with systems that help create an inviting and seemingly safe environment for consumers. Furthermore, these platforms have various methods of generating revenue, which in many cases are also in favor of the vendors' self interests, a fact contributing to their constant success. In this article we document these mechanisms, and investigate their role in the trading ecosystem. Systematically exploring marketplaces, vendor shops, and forums, provides insight on the factors that are contributing the most in shaping the state of the market. We argue that trust plays a vital role in that regard. The reputation that surrounds each vendor, is directly related to the number of clients that are going to decide to purchase their products. Higher reputation translates into more sales, which creates more revenue for the marketplaces hosting the vendors, through purchase commission fees. Taking the trust variable out of the equation, is bound to greatly impact the vendors' profit generation, with cyber attack related products and services as the main focal point. We believe that reputation is one of the foundations of darkweb trading, and hope that this work will inspire more research towards this topic.

## REFERENCES

- [1] V. James King. Here's a breakdown of the \$1.2 billion in silk road drug transactions. [Online]. Available: <https://www.businessinsider.com/heres-a-breakdown-of-the-12-billion-silk-road-drug-transactions-2015-5?r=US&IR=T>
- [2] D. Georgoulas, J. M. Pedersen, M. Falch, and E. Vasilomanolakis, "Covid-19 vaccination certificates in the darkweb," 2021.
- [3] Insights. Geographic distinctions in darknet market activity: U.s. and western europe have the most vendors, eastern europe and china lead in money laundering. [Online]. Available: <https://blog.chainalysis.com/reports/darknet-markets-2021-geographic-breakdown>
- [4] J. Martin and N. Christin, "Ethics in cryptomarket research," *International Journal of Drug Policy*, vol. 35, pp. 84–91, 2016.
- [5] J. Martin, *Drugs on the dark net: How cryptomarkets are transforming the global trade in illicit drugs*. Springer, 2014.
- [6] J. Martin, "Lost on the silk road: Online drug distribution and the 'cryptomarket,'" *Criminology & Criminal Justice*, vol. 14, no. 3, pp. 351–367, 2014.
- [7] D. Décaray-Hétu and J. Aldridge, "Sifting through the net: Monitoring of online offenders by researchers," *European Review of Organised Crime*, vol. 2, no. 2, pp. 122–141, 2015.
- [8] M. J. Barratt, S. Lenton, and M. Allen, "Internet content regulation, public drug websites and the growth in hidden internet services," *Drugs: education, prevention and policy*, vol. 20, no. 3, pp. 195–202, 2013.
- [9] J. Buxton and T. Bingham, "The rise and challenge of dark net drug markets," *Policy brief*, vol. 7, pp. 1–24, 2015.
- [10] B. Collier, D. R. Thomas, R. Clayton, and A. Hutchings, "Booting the booters: Evaluating the effects of police interventions in the market for denial-of-service attacks," in *Proceedings of the internet measurement conference*, 2019, pp. 50–64.

- [11] N. Moretto. Two-factor authentication with totp. [Online]. Available: <https://medium.com/@nicola88/two-factor-authentication-with-totp-ccc5f828b6df>
- [12] T. S. Hyslip, "Cybercrime-as-a-service operations," *The Palgrave Handbook of International Cybercrime and Cyberdeviance*, pp. 815–846, 2020.
- [13] K. Soska and N. Christin, "Measuring the longitudinal evolution of the online anonymous marketplace ecosystem," in *24th USENIX security symposium (USENIX security 15)*, 2015, pp. 33–48.
- [14] E. Nunes, A. Diab, A. Gunn, E. Marin, V. Mishra, V. Palith, J. Robertson, J. Shakarian, A. Thart, and P. Shakarian, "Darknet and deepnet mining for proactive cybersecurity threat intelligence," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2016, pp. 7–12.
- [15] A. Bracci, M. Nadini, M. Aliapoulios, I. Gray, D. McCoy, A. Teytelboym, A. Gallo, and A. Baronchelli, "Dark web marketplaces and covid-19: The vaccines," *Available at SSRN 3783216*, 2021.
- [16] Monero. (2021) Fungibility. [Online]. Available: <https://www.getmonero.org/resources/moneropedia/fungibility.html>
- [17] M. Pechman. (2021) What are bitcoin mixers, and why do exchanges ban them? [Online]. Available: <https://cointelegraph.com/news/what-are-bitcoin-mixers-and-why-do-exchanges-ban-them>
- [18] I. Allison. (2021) Bitcoin tumbler: The business of covering tracks in the world of cryptocurrency laundering. [Online]. Available: <https://www.ibtimes.co.uk/bitcoin-tumbler-business-covering-tracks-world-cryptocurrency-laundering-1487480>
- [19] Monero. (2017) The merits of monero: Why monero vs bitcoin. [Online]. Available: <https://www.monero.how/why-monero-vs-bitcoin>
- [20] Z. Albeniz. (2019) A europol officer confessed that they could not track monero (xmr) transactions. [Online]. Available: <https://medium.com/@ziyahanalbeniz/a-europol-officer-confessed-that-they-could-not-track-monero-xmr-transactions-dbd568f02922>
- [21] F. Harris. Qiwi wallet: An e-wallet payment method! [Online]. Available: <https://cryptomojo.com/qiwi-wallet/>
- [22] A. M. Antonopoulos. Bitcoin q&a: How do i secure my bitcoin? [Online]. Available: [https://www.youtube.com/watch?v=vtzXEs61U&t=0s&ab\\_channel=aantonop](https://www.youtube.com/watch?v=vtzXEs61U&t=0s&ab_channel=aantonop)
- [23] L. Sun. (2020) Blockchain explained: Custodial vs non-custodial wallets. [Online]. Available: <https://medium.com/mogulproductions/blockchain-explained-custodial-vs-non-custodial-wallets-76e6128834b0>
- [24] J. Redman. Sources say world's largest darknet empire market exit scammed, \$30 million in bitcoin stolen. [Online]. Available: <https://news.bitcoin.com/sources-say-worlds-largest-darknet-empire-market-exit-scammed-30-million-in-bitcoin-stolen/>
- [25] J. Aldridge and R. Askew, "Delivery dilemmas: How drug cryptomarket users identify and seek to reduce their risk of detection by law enforcement," *International Journal of Drug Policy*, vol. 41, pp. 101–109, 2017.
- [26] N. Vorobyov. (2020) A new breed of drug dealer has turned buying drugs into a treasure hunt. [Online]. Available: <https://www.vice.com/en/article/g5x3zj/hydra-russia-drug-cartel-dark-web>
- [27] D. W. Link. (2020) How to sell drugs on darknet using dead drops. [Online]. Available: <https://darkweblink.com/sell-drugs-online-dead-drops/#How-To-Format-Dead-Drop-Location>
- [28] D. Rhumorbarbe, L. Staehli, J. Broséus, Q. Rossi, and P. Esseiva, "Buying drugs on a darknet market: A better deal? studying the online illicit drug market through the analysis of digital, physical and chemical data," *Forensic science international*, vol. 267, pp. 173–182, 2016.
- [29] R. Broadhurst, J. Foye, C. Jiang, and M. Ball, "Illicit firearms and other weapons on darknet markets," *Trends and Issues in Crime and Criminal Justice [electronic resource]*, no. 622, pp. 1–20, 2021.
- [30] C. Copeland, M. Wallin, and T. J. Holt, "Assessing the practices and products of darkweb firearm vendors," *Deviant Behavior*, vol. 41, no. 8, pp. 949–968, 2020.
- [31] A. Guirakho. (2019) Understanding the different cybercriminal platforms: Avcs, marketplaces, and forums. [Online]. Available: <https://www.digitalshadows.com/blog-and-research/understanding-the-different-cybercriminal-platforms-avcs-marketplaces-and-forums/>
- [32] N. Christin, "Traveling the silk road: A measurement analysis of a large anonymous online marketplace," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 213–224.
- [33] E. Kermitis, D. Kavallieros, D. Myttas, E. Lissaris, and G. Giataganas, "Dark web markets," in *Dark Web Investigation*. Springer, 2021, pp. 85–118.
- [34] J. Martin, R. Munksgaard, R. Coomber, J. Demant, and M. J. Barratt, "Selling drugs on darkweb cryptomarkets: differentiated pathways, risks and rewards," *The British Journal of Criminology*, vol. 60, no. 3, pp. 559–578, 2020.
- [35] J. Demant, R. Munksgaard, and E. Houborg, "Personal use, social supply or redistribution? cryptomarket demand on silk road 2 and agora," *Trends in Organized Crime*, vol. 21, no. 1, pp. 42–61, 2018.
- [36] G. P. Paoli, J. Aldridge, R. Nathan, and R. Warnes, "Behind the curtain: The illicit trade of firearms, explosives and ammunition on the dark web," 2017.
- [37] S. Pastrana, D. R. Thomas, A. Hutchings, and R. Clayton, "Crimebb: Enabling cybercrime research on underground forums at scale," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1845–1854.
- [38] A. Hutchings and T. J. Holt, "A crime script analysis of the online stolen data market," *British Journal of Criminology*, vol. 55, no. 3, pp. 596–614, 2015.
- [39] T. J. Holt and E. Lampke, "Exploring stolen data markets online: products and market forces," *Criminal Justice Studies*, vol. 23, no. 1, pp. 33–50, 2010.
- [40] A. V. Vu, J. Hughes, I. Pete, B. Collier, Y. T. Chua, I. Shumailov, and A. Hutchings, "Turning up the dial: the evolution of a cybercrime market through set-up, stable, and covid-19 eras," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 551–566.

## APPENDIX

### A. Forum Discussion on Delivery Methods

In this appendix we present some discussions found on the *Dread* forum, regarding the usage of PO boxes.

*"Your name/address is public information, anyone can order something to your house under your name. Your PO box is private as hell. Just one reason I'd prefer a mailbox."* - Dread forum user

*"A lot of vendors will refuse to ship to PO boxes for good reason. If your parcel is sitting at an office stinking of drugs its probably not a good thing. Not to mention a PO box is directly linked to you, where as your address as dumb as this sounds you have deniability as anyone can send anyone a parcel, there is nothing stopping me posting my neighbor a brick of coke however if i was to ship it to their private PO box and it gets found you are going to need a really good lawyer to get out of that one."* - Dread forum user

### B. Marketplace Specific Features

Appendix B is dedicated to illustrating features that have been implemented by marketplaces, with the purpose of keeping the users engaged and entertained, while also serving as an additional source of income for the platform owners.

*a) Deadpool:* This mechanism is deployed by the *Archetype* marketplace and is in essence a betting function. Users can vote on whether each one of the currently active marketplaces, is going to exit scam, retire, or get taken down by law enforcement. The total amount of bets placed is gathered into a pot, which the users with the correct votes win.

*b) Lottery:* *Cartel Marketplace* has implemented a weekly lottery feature. Users can buy tickets for \$1 each, and will be given a unique code at the moment of purchase. At the end of the week a random winning ticket is chosen, and the

entire lottery pool is credited to the winner’s account balance, after a 10% fee is deducted by the marketplace. To reassure the users that the process is fair, there is an additional mechanism in place, which aims to provide transparency. A random seed is published at the start of each week, which along with the winner’s information, winning ticket code and seed, are added onto a blockchain, available for download by all users.

*c) Roulette:* The roulette function from the *Hydra* market, as they explicitly mention on their platform, is intended to “to popularize the HYDRA platform, to attract customers, an increase in the number of orders from stores”. It is implemented as a payment method, where instead of paying directly for the full price of the product, users have the option to take a gamble. They can buy chips which cost around 1% of the product’s price, plus a small added percentage as commission for the marketplace. Each of these chips correspond to 1% of winning probability, so the more chips they buy, the higher the probability to win. They then place the chips on the number they wish from 1-100, and the game starts. There is only 1 winning number chosen each time, which is the integer part of a decimal number with 16 decimal digits, and if it is one of the numbers chosen by the user, they win. In this case the product is automatically bought for the user. Additionally, in the beginning of the lottery, the winning number in its decimal form along with the identifying number of the current lottery, are both hashed and given to the user in order for them to authenticate the result of the lottery.

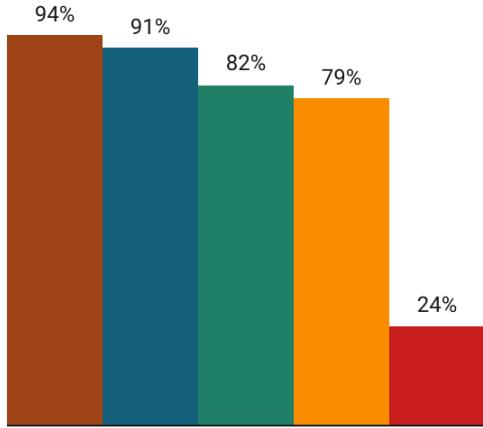


Fig. 3. Mechanism implementation on the 33 marketplaces in percentages. The legend presents the total number of marketplaces that implement each mechanism.

### C. Mechanism Implementation per Marketplace

In this section we present statistics on the usage of user 2FA, CAPTCHA, and marketplace authentication mechanisms, multisignature payment scheme availability, as well as whether these platforms utilize a registration/login “wall” that the users need to bypass before reaching the listing section. It should be specified that this data presented on Figure 3 was collected in a

subsequent phase of writing this paper, which in combination with the dynamic availability of darkweb marketplaces and their short life span, led to only 33 of the initial 41 being fully operational. Many of these marketplaces are still very likely to come online in the future, so we also refrain from providing their names to avoid directing traffic towards them, as discussed in section II-A.

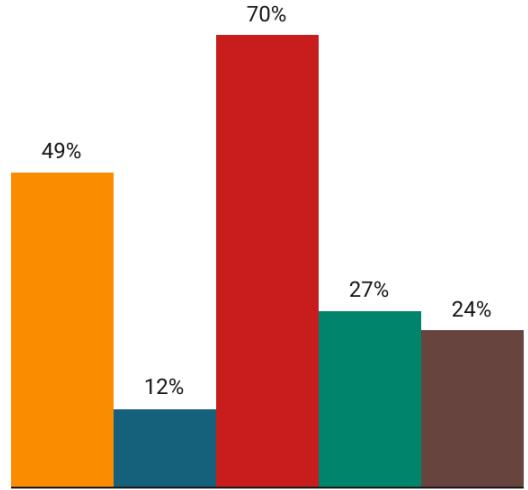


Fig. 4. CAPTCHA type usage on the 33 marketplaces in percentages. The legend presents the total number of marketplaces that implement each CAPTCHA type.

### D. CAPTCHA Types

Lastly, this section is dedicated to the various CAPTCHA types that we discovered while exploring the 41 marketplaces. In common with the information provided in the previous section, the data showcased in this section are collected from 33 out of the total 41 platforms. We documented 4 different families of CAPTCHA mechanisms, namely text-based, mechanisms containing image-based puzzles, question and answer format (e.g. mathematical equation solving), and implementation of the clock CAPTCHA illustrated on Figure 1 (see Section III-A2).

# **Measuring the accuracy of LSTM and BiLSTM models in the application of artificial intelligence by applying chatbot programme**

**Prasnurzaki Anki<sup>1</sup>, Alhadi Bustamam<sup>2</sup>**

<sup>1,2</sup>Department of Mathematics, Universitas Indonesia, Indonesia

<sup>2</sup>Data Science Centre, Universitas Indonesia, Indonesia

---

## **Article Info**

### **Article history:**

Received Dec 30, 2020

Revised May 25, 2021

Accepted Jun 4, 2021

---

### **Keywords:**

Artificial intelligence

BiLSTM

Chatbot

Data science

LSTM

---

## **ABSTRACT**

Python programme contains a question and answer system that derived from data sets that have used and implemented the chatbot in this modern era. where the data collected is in the form of corpuses containing extensive metadata-rich fictional conversations derived from extracted film scripts, commonly called cornell movie dialogue corpus. The various models have been used chatbots in python programmes, and LSTM and BiLSTM models were specifically used in this study. Where the form of accuracy will be reported as a result of the implementation of LSTM and BiLSTM models in the chatbot programme. The programme performance will be influenced by the data from the model selection, because the level of accuracy is determined by the target programme being taken. So this is the main factor that determines which model to choose. Based on considerations required for choosing the programme model, in the end the LSTM and the BiLSTM models are chosen and will be applied to the programme. Based on the LSTM and BiLSTM chatbot programmes that have been tested, it can be concluded that the best parameters come from a pair of BiLSTM chatbots using the BiLTSM model with an average accuracy value of 0.995217.

*This is an open access article under the [CC BY-SA](#) license.*



---

## **Corresponding Author:**

Alhadi Bustamam

Gedung D, Kampus Baru FMIFA, Universitas Indonesia

Depok, Jawa Barat 16424, Indonesia

Email: alhadi@sci.ui.ac.id

---

## **1. INTRODUCTION**

Chatbots are automated systems created to help users by answering their questions. For businesses, chatbots can provide a better way to connect with their customers and increase customer satisfaction levels. Customers get a better, more convenient way to get answers to their questions without waiting on the phone or sending frequent emails [1]. Artificial intelligence (AI) has made an impact in everyday activities by designing and providing evaluation of sophisticated applications and devices, which can perform various functions. Chatbot is an artificial intelligence programme, which is based on the development of AI, it is hoped that the chatbot's ability to imitate human agents in conversation. Chatbots have become so common in their presence that they can reduce service costs and can handle multiple customers simultaneously [2]. Hopefully, future chatbots can improve business sector performance by increasing customer satisfaction levels by saving time. They will also save customer service employees time; customers can use chatbots to get information that previously required humans to answer questions manually.

A fetch model contains several forms based on matches derived from user input and the chatbot can generate answers based on the forms that the user has filled in. Here knowledge used in chatbots is a form of

human hand code. Chatbot knowledge construction is time-consuming and difficult. Therefore, it is very important to have an automated knowledge extraction mechanism to build various forms of chatbots [3]. The use of models that can improve chatbot performance in answering questions automatically can be considered to compare which two models will influence chatbot performance and determine which model has a better fit.

A system that can receive feedback and respond from users and can keep the conversation going, is called a chatbot. The encoder-decoder architecture is used in building parts of the chatbot [4]. A chatbot is a simple robot that contains a programme to answer questions from users. After that, the answer data will be generated from the questions asked by the user. The semantic question-answering system has developed in which words that are uncertain are the form of the question [5]. Application of question and answer system in the form of a chatbot is expected to answer these challenges.

Artificial intelligence (AI) which is the latest technological advancement is very helpful in the development of new virtual assistants to be efficient (online chatbots). Meanwhile, the study also analyzes how existing technological advances made on new chatbots have an impact on future customer support. Going forward, technological innovations in AI allow chatbots to perform increasingly complex tasks [6]. NLP (natural language processing) is a mechanism that can be used to support computer machines by simulating human abilities that function to understand language [7]. Natural language processing is another area where the stance of deep learning can have a huge impact on experimentation that could occur over the next few years [8]. In NLP models, LSTM considers the order dependence between word sequences that the test will perform on the programme to capture dependencies in both the long and short-range forms. BiLSTM can perform both directional scans, allowing simultaneous access to both contexts in forward and backward directions. Therefore, BiLSTM can solve sequence model tasks with better performance than LSTM [9]. Based on the study of these references, this journal will determine whether the BiLSTM model will perform better than the LSTM model in use in NLP.

The discussion conducted on several chatbot backgrounds indicates that consumers' problems, in general, can be presented through several recorded questions that relate to various constraints, such as data storage and limited customer service hours. In order to provide answers given by consumers, a programme is needed to optimize the results of these services. In connection with this, the modeling theory will be discussed in this journal. It is used as the basis when chatbots are deployed in question and answer systems that use Python programmes with LSTM and BiLSTM as models. Then, it is expected that from this research will be seen a comparison between the sentence response generated by the chatbot with the LSTM model and the BiLSTM model with the sentence response in the data set. The solution methodology that will be used in this research is measuring the accuracy of LSTM and BiLSTM models by applying chatbot programme. In more detail, we will start from understanding the background of the importance of the role of the chatbot in the question and answer system, determining the steps for making a chatbot, applying various models, methods, applying data into the programme, write results and discussions, to make conclusions, that have been described in more detail from the session 1 to 6. The major contribution of this paper is to determine the most effective model that can be applied to the chatbot programme based on the comparison of the accuracy results of the two models.

## 2. RESEARCH METHOD

As the various forms of chatbots increasingly integrate the design of AI mechanisms (such as game theory, data mining and optimisation techniques), they comply with these networks' rules and dynamics. This form can be characterised by real multi-actor-based conversations that require technical resources, specialised knowledge and communication skills to maintain online interactions [10]. Summarized by the acronym AI, this is a science that focuses on handling the production of human knowledge, and can offer to the machine the ability to imitate human reasoning and intelligence [11]. AI technology can provide improvements to conversations and collaborate between humans and machines. This technology can be used to create better interactions between humans and machines [12].

The LSTM model, the BiLSTM model and several pairs of parameters, it is also the greedy method can be used in building a chatbot programme by using a set of sentences derived from the data set. The chatbot programme is run based on input in the form of commands from the user, where the results of the programme are a collection of sentences containing information that matches user input based on the relevance of questions and answers.

### 2.1. Steps in making a chatbot

Identifying the identity of the data, inputting the data about answer and question, using the programme of chatbot and then evaluating the output are the 4 steps that must be done when creating a chatbot. First, the identity of the data, cornell movie dialogue corpus is used as data, it contains a collection

of data in the form of a corpus in which it includes a vast collection of fictional conversations rich in meta data extracted from the film script [13]. The data used is from 2018 in the form of text data. Second, in the question and answer system data input contains sentences that come from the film dialogue, then the data functions as user input that is entered into the programme, then after that it is executed into input from the programme user. The third is the chatbot programme development. Fourthly, the following considerations need to be taken to prepare thechatbot programme. A sequence-to-sequence translation will come in several optionss. Selecting the LSTM model will generate the most accurate chatbot response in the end. The final step in the output evaluationis to determine whether the model can provide accurate results or not.

## 2.2. LSTM model

The LSTM model network is known as a model that has had influence in the past and shows the ability to learn from sequential data [14]. The implementation of encoder in this model will be contained in the last hidden statement of the LSTM [4].

$$f_{m+1} = \sigma(\theta^{(h \rightarrow f)} h_m + \theta^{(x \rightarrow f)} x_{m+1} + b_f) \text{ forget gate} \quad (1)$$

$$i_{m+1} = \sigma(\theta^{(h \rightarrow i)} h_m + \theta^{(x \rightarrow i)} x_{m+1} + b_i) \text{ Input gate} \quad (2)$$

$$\tilde{c}_{m+1} = \tanh(\theta^{(h \rightarrow c)} h_m + \theta^{(w \rightarrow c)} x_{m+1}) \text{ update candidate} \quad (3)$$

$$c_{m+1} = f_{m+1} \odot c_m + i_{m+1} \odot \tilde{c}_{m+1} \text{ memory cell update} \quad (4)$$

$$o_{m+1} = \sigma(\theta^{(h \rightarrow o)} h_m + \theta^{(x \rightarrow o)} x_{m+1} + b_o) \text{ Output gate} \quad (5)$$

$$h_{m+1} = o_{m+1} \odot \tanh(c_{m+1}) \text{ Output} \quad (6)$$

The operator  $\odot$  is included in the elemental product. The result of adding hidden states that can be implemented in  $h_m$  with  $c_m$ . as a representation of memory cells is the definition of the LSTM model. The gate form is the value of the memory cells present at each mth time which consists of the sum of the two qualities:  $c_{m-1}$  is the value of the previous memory cell, while  $c_m$  is the value of the memory cell after it has changed, after being calculated from the previous input in the current  $x_m$ form and the previous hidden state in  $h_{m-1}$ form. Furthermore,  $h_m$  is calculated from the cell memory, where the non-linear function path during the update is not passed by the memory cell, so information can be worked on over a remote networ [4]. The preceding hidden state will be determined by each gate controlled by a weighted vector (e.g. $\theta^{(h \rightarrow f)}$ ) and input current (e.g.  $\theta^{(x \rightarrow f)}$ ), plus a vector offset (e.g.  $b_f$ ). The LSTM status after reading tokens is represented ( $h_m, c_m$ ) in an operation that can informally be summarized as  $(h_m, c_m) = \text{LSTM } \llbracket(x_m, (h_{m-1}, c_{m-1}))$ . And the results obtained that LSTM can outperform standard artificial neural networks in a variety of problems displayed in square-shaped gates with dotted edges. the next word  $w_{m+1}$  is stripped using  $h_m$  existing in the LSTM language model. LSTM outperforms standard recurrent neural networks in various problems, such as language modelling problems [4].

One of parallel computerized models is the LSTM model. Parallel computing is a type of computation in which various process calculations can be carried out simultaneously, while the application of parallel computing can run algorithm more quickly in the appearance of the model used in this study [15], [16]. Based on that study, we choose parallel computing models to researched more deeply.

## 2.3. BiLSTM model

The BiLSTM model is a model that combines the advantages of the BiRNN model and the LSTM model [17]. The BiLSTM model is used to propagate the use of forward and reverse directions. The BiLSTM model is a two-way network used to store future data and past data, which is more effective in the LSTM model [18]. In the feature-based model, traits related to shape knowledge are processed by feature suffixes in the neural network. Embeddings are a technique used to handle the sparse matrix of the bag of words. One application of feature suffixes in neural networks is that they can be inserted by constructing the invisible embeddings of words from their spelling or morphology. One way to do this is to incorporate additional two-way RNN layers, one of which is for each word in the vocabulary. The BiLSTM model is one of many parallel computations. The first step is to encode  $w^{(p)}$  dan the  $w^{(q)}$  query using two LSTMs. This process is known as Bidirectional LSTM (BiLSTM)

$$h^{(q)} = BiLSTM(w^{(q)}; \theta^{(q)}) \quad (7)$$

$$h^{(p)} = BiLSTM(w^{(p)}; \theta^{(p)}) \quad (8)$$

The questions are represented by the vector  $u$ , vertically combining final states from left to right, and are represented by matching the ending state vertically from left to right.

$$u = \left[ \overrightarrow{h^{(q)}}_{M(q)}; \overleftarrow{h^{(q)}}_0 \right] \quad (9)$$

Vector  $(u^{(q)})^T$  is the result of applying the vector  $u$  with equation  $h_m = g(x_m, h_{m-1})$ ,  $m = 1, 2, \dots, M$ , (based on [4]) which has been transposed,  $W_a$  is a weight matrix with index  $a$ ,  $h_m^{(p)}$  is the result of implementing the hidden state with (8), and vector  $(\tilde{\alpha}_m)$  is a representation of what is expected and is calculated by,

$$\tilde{\alpha}_m = (u^{(q)})^T W_a h_m^{(p)} \quad (10)$$

$$\alpha = \text{SoftMax}(\tilde{\alpha}) \quad (11)$$

$$o = \sum_{m=1}^M \alpha_m h_m^{(p)} \quad (12)$$

In (11), the vector  $\alpha$  is the result of the SoftMax function of  $\tilde{\alpha}$ . In (12), these vectors can be arranged equal to the corresponding element in  $h^{(p)}$ , assuming that the candidate's answer (vector  $o$ ) is the span of the original text. The score of each candidate for answer  $a$  is calculated by the product in,

$$\hat{c} = \underset{c}{\operatorname{argmax}} o \cdot x_c \quad (13)$$

#### 2.4. Greedy method

The next step after choosing a model is to determine the programme method. The greedy method was chosen as the programme method that is used because it is the implementation of the LSTM model, when it is run, the programme can process data in a faster time, so that it can increase the accuracy of the selected model [19]. While the Greedy algorithm is well understood to be able to produce reasonable estimates for a wide class of functions, it can be seen that it performs much better than one might expect from a greedy algorithm [20]. Based on [18], relatively increasing candidate solutions by trying to approach the optimal solution is Greedy's algorithm. The greedy method is an implementation of the LSTM and BiLSTM models, which is at run time, the time used in processing data is faster and it can increase the accuracy of the selected model.

#### 2.5. Seq2seq model

In Figure 1, the important things that must be understood about the seq2seq model implementation are presented. The seq2seq model functions to generate various responses to user input, so it can implement a question and answer system, where the Python-based Jupyter Notebook Software is chosen as a programme that can view programme input and output [19].

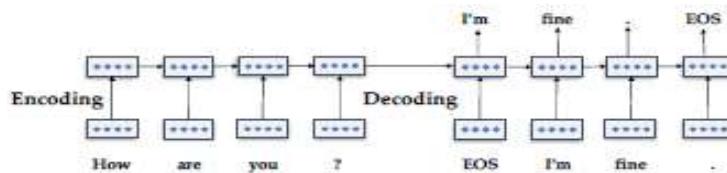


Figure 1. Generation of neural responses in dialogue at the encoder decoder model [19]

### 3. DATA IMPLEMENTATION IN THE PYTHON PROGRAMME

Systematically, the programme planning is compiled in implementing data from the Jupyter Notebook software in the Python programme as follows: Firstly, choose the appropriate software, they are choosing a programme that can process data well, having performance in software data processing and having the availability of supporting attributes that needed in creating the programme. Secondly, the programme performance is influenced by the selection of model that follows the characteristics of the data, so when choosing a model and supporting attributes, it is necessary to pay attention to it. The outcome of a programme

about its ability to determine a high or low level of accuracy are the major factors in choosing which programme to used. So, seeing that the LSTM Model was chosen as the model applied to this programme, it is because it is in accordance with the model selection requirements. In addition to the main factors, supporting factors are also the determining factors whether the processing of data is in line with the criteria in question. It will make the data verifiable. The input sentence that has been processed by the seq2seq model with other programme structures and models, then the output sentence will be issued, it is as a chatbot programme result. Finally, establish the method of programme evaluation. Several choices of evaluation methods that can be used include: loss, accuracy, val\_loss and val\_accuracy, so that from these cases [19] text classification evaluation methods such as simple binary detection tasks can be considered its use.

Spam detection for example, it assigns a positive or negative label to spam according to the category an email document, it should be able to define, and also it must be able to identify an item as spam or not. As presented in (14):

$$\text{accuracy} = \frac{tp+tn}{tp+fp+tn+fn} \quad (14)$$

Based on the theory [20], there are several things that need to be prepared for the value of losses, they include  $x$  for observation and stated is for the function of losses that is expressed by (15):

$$L(\hat{y}, y) = \text{The amount of difference from the true } y \text{ value} \quad (15)$$

Here is a calculation to determine the closeness of the output classification. Here is a calculation to determine the closeness of the output classification  $\hat{y} = \sigma(w \cdot x + b)$  to the actual output ( $y$ , which is 0 or 1). The training process is used to calculate the error rate that is derived in calculated model, it is used to observe the validation set loss function. It is as defined in [21] about the meaning of loss value validation. The model that has been selected and trained, it is then evaluated for its effectiveness as a classification task. This can be done by calculating the percentage of samples that have been classified, as follows:

$$\text{classification rate} = \frac{\text{number of samples correctly classified}}{\text{total number of samples}} \quad (16)$$

The equation (16) shows the misclassification can be calculated and equipped with a classification rate, where is the calculation of the model error rate as follows:

$$\text{error rate} = \frac{\text{number of loss function over validations set}}{\text{total number of validations set}} \quad (17)$$

The loss function calculated on the predefined model validation set will result in the percentage, as was the case of [22], through the graph presented in Figure 2, we can compare the losses the dataset of training and the validation set of val\_loss. From the graph, obtained that the result of a validation loss may be higher or lower than the loss value of the training data set, so this condition is called underfitted or overfitted.

Accuracy is then chosen for the evaluation method of chatbot programme, after comparing each programme's evalution method that accordingly. After getting the suitability value between the form of two sentences, it will be used as the accuracy value that will be used as a chatbot programme evaluation method. The selected model will then be applied to find the differences of words located in sentences during the training period [23].

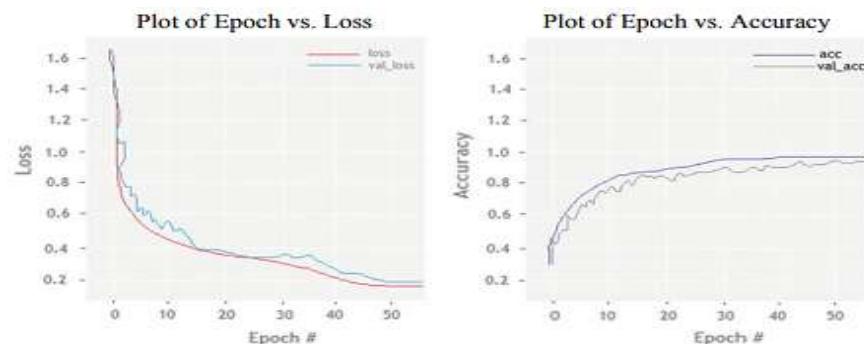


Figure 2. Comparison chart of epoch to loss and epoch to accuracy [22]

#### 4. APPLYING DATA IMPLEMENTATION IN THE PYTHON PROGRAMME

Accuracy was chosen as the method of programme evaluation in accordance with the selected chatbot programme after comparing the evaluation methods of each programme. In order to study the need to implement a chatbot in the question and answer system, this section will explain. The solution methodology that will be used in this research starts from understanding the background of the importance of the role of the chatbot in the question and answer system, determining the steps for making a chatbot, applying various models, methods, applying data into the programme, selecting evaluation methods that have been described in more detail from the session 1 to 3.

##### 4.1. Description of the problem

Many questions will certainly be asked by consumers based on certain data in the face of dynamics in this modern era. In order to provide easy access and convenient operational time, chatbots need to be created. So, the question and answer service between humans and chatbot programmes (machines) can run. This is as an implementation of the question & answer system data.

##### 4.2. Programme making

In Figure 3, we show how to steps by steps in programme making to build the chatbot.

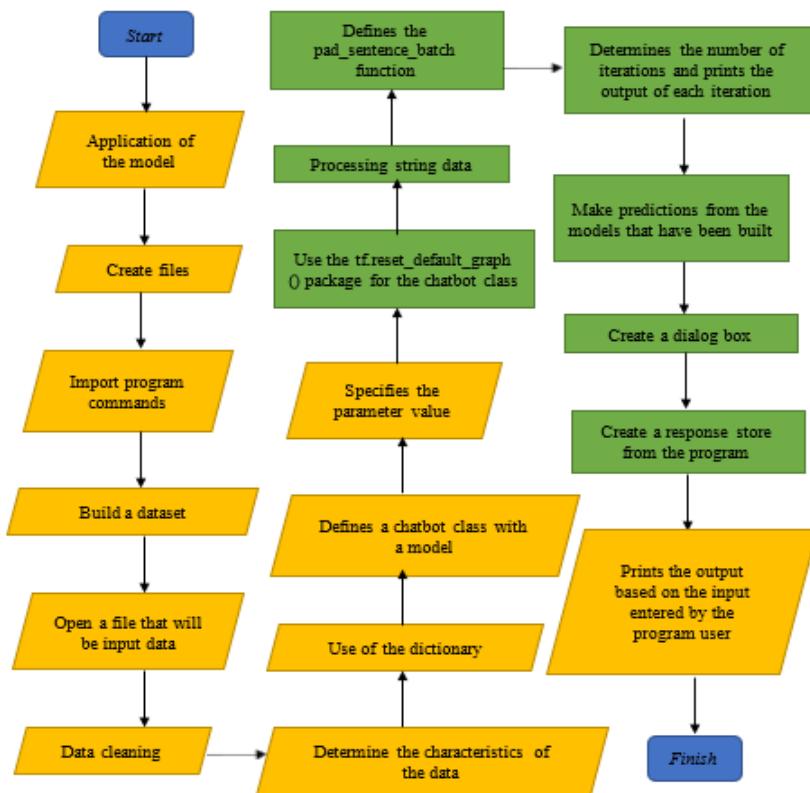


Figure 3. Programme making flowchart

#### 5. RESULTS AND DISCUSSIONS

From the chatbot programme that has been created, 6 files are generated, 3 files from the implementation of the LSTM model into the chatbot programme and 3 files from the BiLSTM model into the chatbot programme. In Table 1, the LSTM model is used to test the parameter pairs. Secondly, epoch (4 different amounts, 20, 30, 40, 50) will be tested with different numbers of epochs, adjusting for other parameter pairs, to determine which parameter pair is the most accurate in the two selected models [24]. In Table 1, the LSTM model is used to test the parameter pairs (File 1, File 2, and File 3), and BiLSTM model (File 4, File 5, and File 6). The purpose of having different values in the parameter is to produce better accuracy and compare the differences between the number of differences in the same parameter, whether it can have better output results than the test results on the parameter pairs in the experiment.

Table 1. Data tested on LSTM and BiLSTM chatbot

Parameter Pair	File 1, File 4	File 2, File 5	File 3, File 6
size_layer	128	256	512
num_layers	2	2	2
embedded_size	64	128	256
learning_rate	0.001, 0.0015	0.001, 0.0015	0.001, 0.0015
batch_size	8	16	32
epoch	20,30,40,50	20,30,40,50	20,30,40,50

According to the number of parameters in LSTM Chatbot, in 3 different files (with each value of the size\_layer parameter different in each file, namely: 128, 256, 512) will be tested for a total of 8 trials each, with 4 parameters with the number of values in the same parameters in each file.

- First, the size\_layer (in the form of the number of layer sizes that will be applied to the programme, in general, multiples of 2 are used such as 16, 32 and 64, based on [25]) in the programme to be discussed is an applied size\_layer with the values 128, 256 and 512.
- Second, Num\_layers (in the form of n number of layers that will be applied to the programme [26]) in the programme to be discussed is an applied size\_layer with a value of 2.
- Third, the embedded\_size (in the form of the number of sizes of the embedded vector that will be applied to the programme); in general, multiples of 2 are used such as 8, 16, 32 and 64 [26] in the programme to be discussed is an applied size\_layer with values 128, 256 and 512.
- Finally, the batch\_size (in the form of a batch size that will be applied) in the programme to be discussed will affect programme performance [24]. Therefore, in this programme, the batch\_size will be tested with a value of 8, 16 and 32.

Besides having the same parameters, it will also be tested for various parameters with a number of values for different parameters. Based on the programme test conducted in [27], one of the parameters tested in a different number of values is epoch. In this programme, many parameters will be tested with a number of different values. The Figure 4 presents the parameter pair data in each file, which is the best parameter pair of the 6 files tested and will not be rated if the value is > 1.0, this condition is called an overfit (where if the training level is very good in accuracy, but when testing the results are not good). In Figure 4 will be selected which has the best average accuracy, which is a scale of 0,0 to 1,0. So, from all the tests that have been done, it is obtained Parameter Pair 1 from file 6 is the best parameter pair of BiLSTM chatbot, which is with an average accuracy value of 0,995217. Based on the reference, on results of applying the BiLSTM model in domain-specific Chinese word segmentation, the accuracy rate is 95.7415% or 0,957415 [17]. Based on comparison result in this study and the results of the application of the BiLSTM model carried out by [17] is better the research reference.

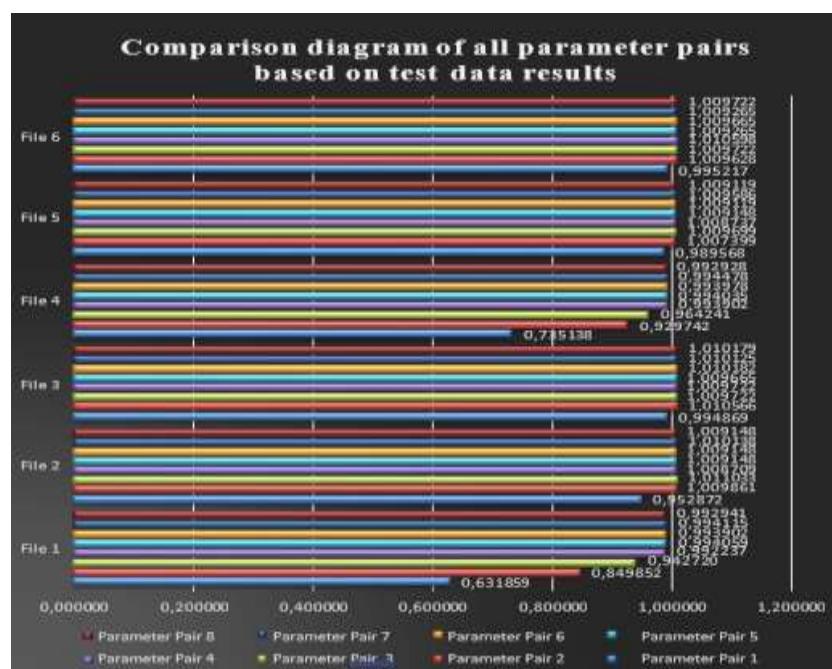


Figure 4. Parameter pair comparison chart based on file data results

Hope in order to achieve more effective result, to get better model performance can combine the models. Other than that, we hope to improve the BiLSTM Chatbot with make modifications to the model architecture that will result in better accuracy [28], [29]. We will also increase the amount of data analysed, aiming to encourage researchers to propose methods that produce better, more efficient results [30].

## 6. CONCLUSION

After applying the BiLSTM model to the chatbot, we were able to deduce from all the test results of the programme that had been conducted with a variety of different parameter pairs, then it is obtained the result, if the Parameter Pair 1 (size\_layer 512, num\_layers 2, embedded\_size 256, learning\_rate 0.001, batch\_size 32, epoch 20) from File 6 is the best parameter pair of BiLSTM Chatbot with an average accuracy value of 0.995217. For future work, the researcher should improve latest model, trying to increase the number of proportions in the data to be studied, so as to produce better research results.

## ACKNOWLEDGEMENTS

This research is partly supported by DRPM research grant 2Q2 2020 with contract number NKB-778/UN2. RST / HKP.05.00 / 2020 from University of Indonesia. The author would like to thank the support from members of the Laboratory of BACL (Bionforatics and Advanced Computing) at the DSC (Department of Mathematics and Data Science) at the Faculty of Mathematics and Natural Sciences, University of Indonesia. Our special thanks to Enago ([www.enago.com](http://www.enago.com)) for the English review of this paper.

## REFERENCES

- [1] M. Nuruzzaman and O. K. Hussain, "IntelliBot A Dialogue-based chatbot for the insurance industry," *Knowledge-Based Systems*, vol. 196, p. 105810, 2020, doi: 10.1016/j.knosys.2020.105810.
- [2] E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Machine Learning with Application*, vol. 2, p. 100006, 2020, doi: 10.1016/j.mlwa.2020.100006.
- [3] S. Arsovski, *et al.*, "Automatic knowledge extraction of any Chatbot from conversation," *Expert Systems With Applications*, vol. 137, pp. 343-348, 2019, doi: 10.1016/j.eswa.2019.07.014.
- [4] J. Eisenstein, "Natural Language Processing," *MIT press*, pp. 137-138, 2018.
- [5] A. C. O. Reddy and K. Madhavi, "Hierarchy based firefly optimized k-means clustering for complex question answering," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 17, no. 1, p. 265, 2020, doi: 10.11591/ijeecs.v17.i1.pp264-272.
- [6] E. Pantano and G. Pizzi, "Forecasting artificial intelligence on online customer assistance: Evidence from chatbot patents analysis," *Journal of Retailing and Consumer Services*, vol. 55, p.102096, 2020, doi: 10.1016/j.jretconser.2020.102096.
- [7] S. J. and S. Swamy, "A prior case study of natural language processing on different domain," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, p. 4928, 2020, doi: 10.11591/ijece.v10i5.pp4928-4936.
- [8] P. Patel and A. Thakkar, "The upsurge of deep learning for computer vision applications," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, p. 547, 2020, doi: 10.11591/ijece.v10i1.pp538-548.
- [9] W. Li, *et al.*, "Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification," *Neurocomputing*, vol. 381, p. 64, 2020, doi: 10.1016/j.neucom.2020.01.006.
- [10] G. Murtarelli, A. Gregory, S. Romenti, "A conversation-based perspective for shaping ethical human-machine interactions: The particular challenge of chatbots," *Journal of Business Research*, vol. 129, pp. 927-935, 2020, doi: 10.1016/j.jbusres.2020.09.018.
- [11] S. Berhil, H. Benlahmar, N. Labani, "A review paper on artificial intelligence at the service of human resources management," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 18, no. 1, p. 35, 2020, doi: 10.11591/ijeecs.v18.i1.pp32-40.
- [12] S. R. Salkuti, "A survey of big data and machine learning," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, p. 2088-8708, 2020, doi: 10.11591/ijece.v10i1.pp575-580.
- [13] R. Chidananda, "Cornell Movie-Dialogs Corpus," *Kaggle datasets*, 2018.
- [14] M. Yaqub, *et al.*, "Modeling of a full-scale sewage treatment plant to predict the nutrientremoval efficiency using a long short-term memory (LSTM) neural network," *Journal of Water Process Engineering*, vol. 37, p. 3, 2020, doi: 10.1016/j.jwpe.2020.101388.
- [15] H. Fang, *et al.*, "An efficient radial basis functions mesh deformation with greedy algorithm based on recurrence Choleskey decomposition and parallel computing Parallel computing and swarm intelligence based artificial intelligence model for multi-step-ahead hydrological time series prediction," *Journal of Computational Physics*, vol. 377, p. 186, 2019, doi: 10.1016/j.jcp.2018.10.029.
- [16] J. M. Sadler, *et al.*, "Leveraging open source software and parallel computing for model predictive control of urban drainage systems using EPA-SWMM5," *Environmental Modelling & Software*, vol. 120, p. 11, 2019, doi: 10.1016/j.envsoft.2019.07.009.

- [17] D. Shao, *et al.*, "Domain-Specific Chinese Word Segmentation Based on Bi-Directional Long-Short Term Memory Model," *IEEE Access*, vol. 7, p. 12996, 2019, doi: 10.1109/ACCESS.2019.2892836.
- [18] I. Attri and M. Dutta, "Bi-Lingual (English, Punjabi) Sarcastic Sentiment Analysis by using Classification Methods," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, p. 1386, 2019.
- [19] D. Jurafsky and J. H. Martin, "Speech and Language Processing An Introduction to Natural Language Processing," *Prentice Hall*, 2020.
- [20] L. Brown and S. Steinerberger, "Positive-Definite Functions, Exponential Sums and the Greedy Algorithm: a Curious Phenomenon," *Journal of Complexity*, vol. 61, p. 9, 2020, doi: 10.1016/j.jco.2020.101485.
- [21] C. Lim, "An Evaluation of Machine Learning Approaches to Natural Language Processing for Legal Text Classification," *Imperial College London*, 2019.
- [22] K. Kotecha, *et al.*, "Data Science and Intelligent Applications: Proceedings of ICDSIA 2020," *Springer Singapore*, vol. 52, p. 117, 2020, doi: 10.1007/978-981-15-4474-3.
- [23] Y. Long, *et al.*, "Repeat Padding: Balancing words and sentence length for language comprehension in visual question answering," *Information Sciences*, vol. 529, pp. 170-171, 2020, doi: 10.1016/j.ins.2020.04.034.
- [24] S. Kalya, M. Kulkarni, and K. S. Shivaprakasha, "Advances in Communication, Signal Processing, VLSI, and Embedded Systems: Select Proceedings of VSPICE 2019," *Springer Nature*, vol. 614, p. 307, 2019, doi: 10.1007/978-981-15-0626-0.
- [25] H. Yin, *et al.*, "Intelligent Data Engineering and Automated Learning – IDEAL 2019, 20th International Conference, Manchester, UK, November 14–16, 2019, Proceedings, Part II," *Springer Nature*, vol. 11872, p. 182, 2019, doi: 10.1007/978-3-030-33617-2.
- [26] L. Barolli, P. Hellinckx, T. Enokido, "Advances on Broad-Band Wireless Computing, Communication and Applications: Proceedings of the 14th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2019)," *Springer Nature*, vol. 97, 2019, doi: 10.1007/978-3-030-33506-9.
- [27] K. Fereday, *et al.*, "A comparison of rolling averages versus discrete time epochs for assessing the worst-case scenario locomotor demands of professionalsoccer match-play," *Journal of Science and Medicine in Sport*, vol. 2, pp. 765-766, 2020, doi: 10.1016/j.jsams.2020.01.002.
- [28] Y. Nie, P. Jiang, H. Zhang, "A novel hybrid model based on combined preprocessing method and advanced optimization algorithm for power load forecasting," *Applied Soft Computing Journal*, vol. 97, pp. 16-17, 2020, doi: 10.1016/j.asoc.2020.106809.
- [29] P. Silitonga, *et al.*, "Comparison of Dengue Predictive Models Developed Using Artificial Neural Network and Discriminant Analysis with Small Dataset," *Appl. Sci.*, vol. 11, p. 15, 2021, doi: 10.3390/app11030943.
- [30] G. Ardanewari, A. Bustamam, D. Sarwinda, "Implementation of plaid model biclustering method on microarray of carcinoma and adenoma tumor gene expression data," *Journal of Physics: Conference Series*, vol. 893, p. 4, 2017, doi: 10.1088/1742-6596/893/1/012046.

## BIOGRAPHIES OF AUTHORS



**Prasnurzaki Anki B.Sc.** received the BSc (honour) degree in mathematics from Universitas Indonesia in 2020. He is currently pursuing a master's degree in mathematics from Universitas Indonesia. His research interests are in the areas of computational mathematics, data science, and artificial intelligence.



**Assoc. Prof. Alhadi Bustamam, Ph.D.** received the BSc (honour) degree in computational mathematics in 1996, the master's degree in computer science from Universitas Indonesia in 2002, and the Ph.D degree in bioinformatics from the University of Queensland, Australia, in 2011. His research focuses on high-performance computing approaches to computational mathematics, computational biology, bioinformatics, computer science, data science, and artificial intelligence. Currently, he is working as an Associate Professor and the Head of Bioinformatics and Advanced Computing Laboratory (BACL) at the Department of Mathematics. He is also serving as the chairman of Data Science Centre (DSC) <https://dsc.ui.ac.id> at Universitas Indonesia.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/356420767>

# The Intelligence Corpus, an Annotated Corpus of Definitions of Intelligence: Annotation, Guidelines, and Student Research Projects

Conference Paper · November 2021

DOI: 10.21125/iceri.2021.0871

---

CITATIONS

0

READS

203

6 authors, including:



Dagmar Monett  
Hochschule für Wirtschaft und Recht Berlin

85 PUBLICATIONS 261 CITATIONS

[SEE PROFILE](#)



Laura Haase  
Hochschule für Wirtschaft und Recht Berlin

2 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)



Marc Normann  
Technische Hochschule Aschaffenburg

1 PUBLICATION 0 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Project Research on Excellent Teaching (RET) [View project](#)



Project HWR-Kompetenzzentrum "Digitalisierung - Auswirkungen auf Geschäftsmodelle und Arbeitswelten" (CC Digital) [View project](#)

# THE INTELLIGENCE CORPUS, AN ANNOTATED CORPUS OF DEFINITIONS OF INTELLIGENCE: ANNOTATION, GUIDELINES, AND STUDENT RESEARCH PROJECTS

D. Monett<sup>1</sup>, L. Hoge<sup>2</sup>, L. Haase<sup>3</sup>, L. Schwarz<sup>4</sup>, M. Normann<sup>5</sup>, L. Scheibe<sup>6</sup>

<sup>1</sup>Berlin School of Economics and Law (GERMANY)

<sup>2</sup>Robert Koch Institute (GERMANY)

<sup>3</sup>Technical University of Applied Sciences Wildau (GERMANY)

<sup>4</sup>Hochschule Stralsund (GERMANY)

<sup>5</sup>NORDAKADEMIE Graduate School (GERMANY)

<sup>6</sup>DB Systel GmbH (GERMANY)

## Abstract

Intelligent systems are transforming the way we interact with technology, with each other, and with ourselves, and knowing at least what artificial intelligence (AI) means is becoming essential for designing, developing, deploying, using, and even regulating intelligent artefacts. Although defining intelligence has been one of the most controversial and studied challenges of both ancient and modern human thinking, a lack of consensus on what intelligence is has remained almost constant over the centuries. We argue that a better understanding of contemporary technologies, AI-based but not only, starts with a grounded exposure to their conceptual pillars. These include fundamental concepts like the concept of intelligence, in general, and of AI, in particular. Learners and decision makers at all levels should face them, as well as be able to discuss their importance and limitations critically and in an informed way. For doing that, they must be confronted with definitions of (artificial) intelligence and understand their meaning well, for instance. If these contents are already part of study programs, the better. In this paper we present how several definitions of intelligence were annotated, i.e. their properties and characteristics systematically analyzed and commented, in order to construct a corpus (i.e. a collection) of definitions of intelligence for further uses in AI and other fields. The work and the concrete application domain presented here have not yet been considered in the extended work on linguistic annotation (i.e. annotating definitions). Even though, both the annotation and the data merit special attention, for they deal with the elusive, important concept of intelligence, i.e. with definitions of both human and machine (or artificial) intelligence. Undergraduate Computer Science students carried out the annotation process and other related research activities. They were involved in a more general AI research project and included their findings and work as part of their undergraduate student research projects in their last study year. We provide details about how the student research projects were conceived, conducted, and mentored.

Keywords: AI literacy, annotation, artificial intelligence, corpus, intelligence, student research projects.

## 1 INTRODUCTION

A lack of consensus on defining intelligence has been a shaky stepping-stone not only for the artificial intelligence (AI) community: interested scholars have not come up with a cross-domain accepted definition of intelligence. Neither in the ancient Eastern nor in the ancient and contemporary Western conceptions of intelligence (see e.g. [1], [2], [3]) nor in the more recent perspectives from the last 70 years within the field of AI (see e.g. [4], [5], [6]).

There are several underlying reasons for disagreement on defining intelligence whose analysis would be beyond the scope of this paper (we refer the interested reader to [7] and [8] for related discussions on the lack of consensus). In Hunt and Jaeggi's [7] words, "*[I]t is not surprising that defining the subject matter of intelligence research has been difficult, for in everyday discourse the word intelligence is used in various ways.*" Dickson [9] emphasizes that the definition of (artificial) intelligence "shifts with technological advances and our expectations from computers. That's why it's pretty hard to determine what is or isn't AI." And Chollet [10] states that "*[T]o make progress towards the promise of [the AI] field, we need precise, quantitative definitions and measures of intelligence—in particular human-like general intelligence.*" Furthermore, the pressing need for clearer, good definitions of intelligence has crossed the academic river, reaching the industry, law, and public shores in unprecedented ways.

Delineating the boundaries of the discourse on intelligence may help in defining and understanding its most discussed concept, as suggested in [11]. Furthermore, better insights into definitions and how to define them has proven to be essential for a better understanding of concepts, intelligence and AI included (see for example [12], [13] and [14] for more on properties of good definitions). Knowing those concepts and related cognitive abilities (like defining, analyzing, understanding, discussing, and comparing definitions of intelligence, among others) is expected for AI researchers and practitioners in the first place. Yet, they are also central to extending or at least providing the basics of *AI literacy* to other stakeholders of our society.

It is the main goal of this paper to present how a few hundreds of definitions of intelligence (of both human intelligence and machine intelligence) were annotated by taking into account different properties of good definitions. In doing so, we follow the guidelines for annotation case studies suggested in [15], which also guide the structure of the paper and our methodology in what follows.

## 2 ANNOTATING DEFINITIONS OF INTELLIGENCE

The annotation case study that is the focus of this paper belongs to a rather uncommon domain in linguistic annotation: definitions of human and machine (or artificial) intelligence are annotated according to quality criteria for definitions. In other words, properties of good definitions are evaluated in order to conclude whether a certain definition of intelligence fulfils these properties or not. To our knowledge, this is the first time that such a problem is tackled in the sub-field of linguistic annotation. Next sections will provide the background and characteristics of this atypical annotation project.

### 2.1 The Annotators

The annotation of data either its nature can be a very challenging and time consuming process. On the one hand, it is a repetitive task fundamentally done by humans (i.e. annotators), mainly because the state of the art in automatic data annotation is still biased, error prone, and far from being entirely satisfactory. On the other hand, data is labelled according to its characteristics, but, even when done by humans, the annotation itself might require special insights into the problem domain. Furthermore, it might need a certain level of agreement on how to interpret and annotate the data correctly, as well as depend on advanced domain knowledge.

Software solutions are available for supporting annotators in their work (see e.g. an extensive review in [16]), but not for all kinds of data and certainly not for all kinds of situations that require specialized knowledge for annotating the data. This is the case when annotating definitions of intelligence according to several quality criteria, where AI-related knowledge might be critical and, thus, a pre-requisite for annotating.

In the case of our annotation project, undergraduate Computer Science students in their third-year studies are the annotators, the majority of them also attending a parallel course on AI. Furthermore, they were involved in related research tasks and completed corresponding student research projects that were especially considered as part of their term evaluation. This way, they could include the knowledge and practice they acquired by annotating the data into their learning and study, directly.

### 2.2 The Annotation Data

The annotation corpus consists of four collections of definitions of intelligence. Participants to a survey on definitions of intelligence [17] were asked to provide their level of agreement with definitions of both human and machine intelligence from the literature (for more on the survey, please consult the provided reference). They were also asked to justify their selection, as well as to provide new definitions of intelligence, if desired. A total of 567 responses from experts worldwide were received and contained more than 4000 comments or arguments in favor or against the literature definitions that were presented to them. Respondents also provided more than 300 new, suggested definitions of intelligence (213 definitions of human intelligence and 125 definitions of machine intelligence). This is how a mixed pool of what experts in other domains call “implicit theories” of intelligence (or people’s conceptions or what intelligence is) and “explicit theories” of intelligence (i.e. theories proposed by experts) was created (see [3] for more on implicit and explicit theories).

Tab. 1 shows the information contained in each collection. The four collections conform what we call *the Intelligence Corpus*.

Table 1. The Intelligence Corpus.

Collection	Content	Definitions
A	New, suggested definitions of <i>machine</i> or artificial intelligence by participants to the survey on defining intelligence [17].	213
B	New, suggested definitions of <i>human</i> intelligence by participants to the survey on defining intelligence [17].	125
C	Definitions of intelligence <i>from the literature</i> to agree upon in the initial edition of the survey on defining intelligence [17].	34
D	Definitions of intelligence from the collection presented in [18].	71

The following examples give an idea of the kind of definitions that are part of the Intelligence Corpus:

*"Machine Intelligence is concerned with building systems that can adapt and learn in unstructured noisy domains."* (From collection A)

*"[Human intelligence is] the ability to use information to accomplish goals."* (From collection B)

*"Intelligence measures an agent's ability to achieve goals in a wide range of environments."* (From collection C)

*"[Intelligence is] the capacity to learn, reason, and understand."* (From collection D)

As it can be seen, and compared to other case studies in linguistic annotation, the Intelligence Corpus is very small. Actually, it is very unlikely (indeed, not expected at all) that considerably many new definitions of intelligence are defined by experts and non-experts alike in a long-term future.

### 2.3 The Annotation Scheme

The annotation scheme referred to in this paper builds upon different works on properties of good definitions some of which were referenced to in Section 1. It uses most of the properties or quality criteria for definitions suggested in [14], which includes a compendium and thorough analysis of the literature on definitions together with their most desirable properties.

The following examples give an idea of the kind of quality criteria that were considered when annotating the aforementioned definitions:

*A good definition of intelligence defines the "what," the thing to be defined. It defines [machine | artificial | human] intelligence.*

*A good definition of intelligence is affirmative.*

*A good definition of intelligence is comprehensive, in that it omits no essential attribute of the thing to be defined; it omits nothing which is a part of [machine | artificial | human] intelligence.*

*A good definition of intelligence is clear, in that it avoids metaphorical, ambiguous language, and obscure terms. It is clearly written; it is perspicuous.*

Notice that some quality criteria are intuitive and easy to understand (and, thus, to verify), whereas others might be more complex, could require a deeper understanding (and, consequently, a thorough evaluation) as well as corresponding added efforts and time for assessing whether a certain definition fulfills the quality criteria or not.

From the 30 quality criteria for definitions introduced in [14], 21 were considered for annotating each definition from the Intelligence Corpus.

### 2.4 The Physical Representation

The collections from Tab. 1 were available in the form of MS Excel tables, one definition of intelligence per row. It was both a logical and straightforward step to extend them with new columns, each representing a property or quality criterion. The new tables were then imported into Google Sheets and prepared to make them available to the annotators, i.e. to the students, in a later step.

Because of the characteristics of the annotation schema and the size of the Intelligence Corpus, it was not necessary to use any other software or system for annotating. The concrete form and type of the annotated data will be clearer in Section 2.5.1 below.

## 2.5 The Annotation Process

The annotation process was done manually. On the one side, a reliable and consistent automatic or semi-automatic annotation of data for this very specific case study was not (and we do not think it will be in an advisable future) available: human language understanding continues to be an unsolved problem in the field of AI. On the other side, the advantage of having a small corpus did not merit the investment in extra resources that might slow down the annotation process as a whole.

Six annotators were involved, three female and three male, all of them undergraduate students in their third year of Computer Science studies, as introduced above. This allowed for at least a satisfactory level of knowledge about the definition of concepts, in general, and of AI, in particular. Crowdsourcing mechanisms for annotating were discarded: not only the size of the corpus was small, but we also assumed that the high-level subject matter might require an added, special training of the annotators, thus at least some exposure to related fields and topics was a requirement.

Three pairs of annotators were formed. Each pair annotated one third of the definitions from the corpus, i.e. 147 or 148 definitions of intelligence in total for each pair of annotators (see Fig. 1). Each annotator annotated her/his definitions independently.

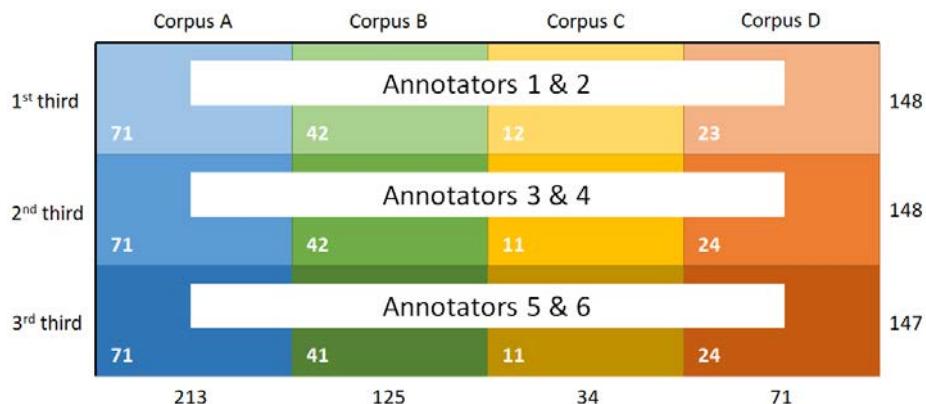


Figure 1. Distribution of definitions per groups of annotators.

The annotators were trained before the annotation process started. An initial meeting was held for this purpose. The training consisted of a general introduction to research topics involving the definition of the concept of intelligence, to the quality criteria for definitions, to related literature including the survey introduced in Section 2.2 (see [17] for more), to the collections of definitions that should be annotated, as well as to the annotation guidelines that will be presented in short. Furthermore, examples of definitions and how to annotate them when considering the quality criteria for definitions were also discussed.

Additionally, all annotators received the same information about the annotation process per email, as well as the annotation guidelines and the URLs with tables in Google Sheets containing both “their” to-be-annotated definitions and the quality criteria to evaluate them. As it was introduced in Section 2.4, the annotation tables contained as many rows as there were definitions of intelligence to be annotated (at most 148 definitions per annotator), and as many additional columns as there were quality criteria to be considered (a total of 21 quality criteria).

Feedback from annotators was collected at the end of the annotation process. The feedback included the time the students spent annotating the definitions of intelligence, which strategies they followed for the annotation, as well as general comments and remarks, if they had any. The annotators sent their results in a period ranging between less than two and up to nine weeks. It worth mentioning that they annotated the definitions and worked on the corresponding student research projects parallel to their attending other learning modules and classes.

### 2.5.1 Annotation Guidelines

Extra, specific to the case study annotation guidelines were especially conceived for the project. They followed some recommendations introduced in [19] and [20]. The guidelines include particular characteristics of as well as some relevant aspects that should be considered when evaluating quality criteria for definitions, together with the activities for doing so. They are listed in what follows in the form that was presented to the annotators:

- **How to proceed:** You can select one column (i.e. one quality criterion) and go row by row (i.e. definition by definition) to evaluate the same criterion for all rows. This could be faster than fixing a row (i.e. fixing one definition) and then analyzing all columns (i.e. all quality criteria) for that row. But you could also go the other way around because some columns are related or refer to similar criteria, plus you need to consider the same definition only once. It is up to you!
- **Write a 1** on a cell if the corresponding definition fulfils the quality criterion on the top of the column. For example, if a definition  $d$  defines machine intelligence (or human intelligence or intelligence, depending on the collection it belongs to) then write a 1 on the cell corresponding to the quality criterion  $d$  defines the “what,” the thing to be defined. Leave the **cell empty** if not.
- Mark a cell in red (i.e. set the **background color** of the cell to red) or write an email asking for clarification, in case you don’t have any idea about how to evaluate a given quality criterion for that cell. Such cases will be discussed in the team later.
- Notice that you **don’t have to justify** your annotation. But, if you prefer, you could use the free columns on the right to write any **comments** or questions related to some particular “difficult case” that needs discussion. This should not be the normal case, though.
- **Annotate alone.** Do not discuss with other annotators about how to annotate a particular definition because this could introduce some bias in yours’ or others’ thinking. If necessary, write an email asking for clarification.
- **Do not fix grammatical errors** you might find in the definitions.
- How long did it take? **Record the time** you spend annotating whenever possible. This will be very useful for the upcoming publication about the annotation process!
- Write an email when you are **finished** with the annotations!
- Got any new **idea or suggestion** that could be included in these guidelines? They are welcome! Drop a line in any case.
- **Extra:** At the end of the annotation (or, better, during the process, if you prefer) write down your “strategy,” i.e. what did you do and how; which problems, difficulties, or positive things did you find, etc. This could be part not only of the research documentation about the annotation process but also of your student research paper later!

As it was already mentioned, these guidelines for annotating definitions of intelligence were also presented and explained to all annotators in the initial meeting.

## 3 RESULTS AND DISCUSSION

This section summarizes the most important results and lessons learned.

### 3.1 Feedback from the Annotators

The time spent on the annotation by each annotator was between 4.5 and 8.5 hours, with an average time of 7 hours. One of the annotators did not record the time and gave as reason the varying conditions under which his annotation sessions took place (at home, at the university, in the train). A second annotator reported having consumed between 8 and 9 hours. In this case, a middle point was considered when calculating the total average time. On average, each annotator invested about three minutes on each definition and more than eight seconds on each quality criterion.

Evaluating whether a definition is *affirmative* or not is easy: for humans, it is straightforward to detect adverbs that denote negation. For example, the definition “[Intelligence is] the capacity to learn, reason, and understand” is posed in an affirmative way, there is even no need to read it until the end. Yet, evaluating whether the same definition is *comprehensive* might require a more complex thinking process. This shows how complicated or time consuming the annotation of a definition could be.

Four annotators reported their individual strategies for annotating. All of them proceeded by fixing a quality criterion and then annotating all definitions according to that criterion. General remarks concerning the annotation process included concrete interpretations of the quality criteria. Such remarks were reported by three annotators.

### 3.2 Inter-Annotator Agreement

The data from the annotators was easy to process once all annotations were available. Before that, the project leader checked the annotations for consistency, randomly.

Then, the inter-annotator agreement (IAA) was computed following Cohen's work [21]. Tab. 2 shows the results for each collection from the Intelligence Corpus and each group of annotators, together with averaged values. This part of the project was the particular research topic and focus of one of the students.

*Table 2. Cohen's  $\kappa$  per collection and group of annotators.*

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>Avg. per group</b>
Annotators 1 and 2	0.390	0.455	0.344	0.411	0.400
Annotators 3 and 4	0.346	0.361	0.457	0.465	0.408
Annotators 5 and 6	0.404	0.431	0.372	0.361	0.392
<b>Avg. per collection</b>	0.380	0.416	0.391	0.412	<b>Absolute: 0.4</b>

The IAA in the same group was between *fair* and *moderate* for all collections (i.e. Cohen's  $\kappa$  ranging from 0.344 to 0.465, and according to Landis and Koch's [22] interpretation of the values).

In general, the number of agreements among annotators was higher for the collection containing definitions of human intelligence, followed by the collection from [18], which includes many dictionary definitions of intelligence that, in general, are clearer and easier to understand. One possible interpretation is that definitions of artificial intelligence, both those provided by participants to the survey and from the literature, are still needing some work regarding expressiveness.

The quality criteria with the highest IAA values were those simpler, more intuitive, and easier to understand, as expected. However, the quality criteria for definitions with the highest number of disagreements (and thus, smaller IAA values among the annotators) were the following ones, in this order:

A good definition of intelligence is *exclusive*, in that it includes nothing which is not a part of [machine | artificial | human] intelligence.

A good definition defines the “*why*,” the purpose of the thing to be defined. It defines the purpose of [machine | artificial | human] intelligence.

In future annotation processes, it might be advisable to abound and explain better to the annotators what certain criteria mean, as well as to use more already (correctly) annotated definitions as examples.

Similarly, it was analyzed which definitions of intelligence received the highest and lowest number of agreements (or disagreements). For example, the annotators were more agreeable when evaluating the fulfilment of the quality criteria for the following definition:

*“Intelligence is a very general mental capability that, among other things, involves the ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience.”*

This result is not surprising: that one is Gottfredson's definition of intelligence [23]. Gottfredson's is not only a widely accepted definition of intelligence among experts in intelligence and allied fields [24], but it was also the most accepted definition of human intelligence in the survey presented in [17]. The annotators confirmed once again what a well-defined definition of intelligence looks like.

### 3.3 Usage

Both the annotated corpus and the original collections of definitions of intelligence (see Tab. 1, Section 2.2) are available upon request. They could be used by interested readers and practitioners, for

instance, when learning about fundamental concepts like the concept of intelligence, in general, and of AI, in particular.

As an example, we provide part of the Intelligence Corpus as a separate collection with 148 definitions of intelligence that were annotated by one of the students. It can be found at <https://bit.ly/AnnotatedDefsIntelligence> (see [25]) under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license. It contains the following information:

- 71 definitions of machine or artificial intelligence (from a total of 213) from collection A,
- 42 definitions of human intelligence (from a total of 125) from collection B,
- 12 definitions of intelligence (from a total of 34) from collection C, and
- 23 definitions of intelligence (from a total of 71) from collection D,

together with their annotations, i.e., whether they fulfill 21 quality criteria for definitions (see Section 2.3).

Furthermore, all definitions considered in the survey on defining (machine) intelligence [17] are available at <https://goo.gl/KDPtKT>, including their complete bibliographic information.

Finally, there is also an app that was developed by the project leader (also supervisor of the student research projects) for the purpose of supporting end users through the process of defining a definition. For example, all quality criteria for definitions are specified and exemplified there. The *Definitely app*, as it is called, may also assist future annotators in their annotation processes (visit <https://definitely.glideapp.io/> for more).

### 3.4 Mentoring

There was enough prior experience available in mentoring and supervising student research projects of the kind presented in this paper. Further, the course on AI was delivered by the same instructor, what allowed not only for *ad hoc* discussions about the state of the art of the mentioned student research projects, but also about their content and goals in a broader setting. Other students also attended the AI course, thereby enriching their general knowledge and projects they were working on. Further, all necessary information not only key to starting the annotation process, but also the one concerning both project management and mentoring was carefully prepared, discussed with, and used by the students effectively.

## 4 CONCLUSIONS

This paper presented an annotated corpus of definitions of intelligence, the Intelligence Corpus, as well as details about its annotation, which was performed as part of student research projects in Computer Science. The Intelligence Corpus forms part of a peculiar annotation case study that evaluates whether definitions of human and machine intelligence satisfy desirable properties or quality criteria of good definitions. Future work includes a thorough discussion about some of the quality criteria (like those more difficult to interpret or annotate) and how to ease further annotation processes. Furthermore, a detailed, manually-conducted quality control of all available annotations will be performed in a near future. Occasionally, the corpus may be extended with new annotated definitions and/or new quality criteria.

Other possible uses of the Intelligence Corpus include training on the process of defining a good definition of any concept, which could be of interest to regulators or lawyers, for instance. In their case, it is essential to deal with legal definitions of different terms and, some of the times, they should even define the definitions themselves. Examples from the Intelligence Corpus would illustrate desirable properties for good definitions and help them in their work. In a similar vein, the Intelligence Corpus could be a complement to students, in particular, and academics, in general, that are learning how to conduct (or that are actually conducting) a concept analysis [26], like Philosophy students, for instance. Last, but not least, further uses of the corpus involving machine learning techniques to analyze its content are not discarded.

## REFERENCES

- [1] H.O. Rugg, “Intelligence and Its Measurement: A Symposium,” *Journal of Educational Psychology*, vol. 12, pp. 123–147, 1921.

- [2] R.J. Sternberg and D.K. Detterman, “*What is Intelligence?: Contemporary Viewpoints on its Nature and Definition*,” Ablex Publishing Corporation, Norwood, NJ., 1986.
- [3] S.-Y. Yang and R.J. Sternberg, “Conceptions of intelligence in Ancient Chinese Philosophy,” *Journal of Theoretical and Philosophical Psychology*, vol. 17, pp. 101–119, 1997.
- [4] D. Monett, C.W.P. Lewis, and K.R. Thórisson, “Introduction to the JAGI Special Issue ‘On Defining Artificial Intelligence’—Commentaries and Author’s Response,” *Journal of Artificial General Intelligence*, vol. 11, pp. 1–4, 2020.
- [5] N.J. Nilsson, “*The Quest for Artificial Intelligence: A History of Ideas and Achievements*,” Cambridge University Press, 2010.
- [6] P. Wang, “On Defining Artificial Intelligence,” *Journal of Artificial General Intelligence*, vol. 10, no. 2, pp. 1–37, 2019.
- [7] E. Hunt and S.M. Jaeggi, “Challenges for Research on Intelligence,” *Journal of Intelligence*, vol. 1, pp. 36–54, 2013.
- [8] D. Monett, L. Hoge, and C.W.P. Lewis, “Cognitive Biases Undermine Consensus on Definitions of Intelligence and Limit Understanding,” in *Joint Proceedings of the IJCAI-2019 Workshops on Linguistic and Cognitive Approaches to Dialog Agents and on Bridging the Gap Between Human and Automated Reasoning* (U. Furbach, S. Hölldobler, M. Ragni, R. Rzepka, C. Schon, J. Vallverdu, and A. Włodarczyk, eds.), pp. 51–58, Macau, China. CEUR-WS, 2019.
- [9] B. Dickson, “5 european companies that are (really) advancing AI,” The Next Web, 2019. Retrieved from <https://thenextweb.com/artificial-intelligence/2019/03/29/5-european-companies-advancing-ai/>.
- [10] F. Chollet, “*The Measure of Intelligence*,” arXiv e-prints, arXiv:1911.01547 [cs.AI], 2019.
- [11] D. Monett and C. Winkler, “Using AI to Understand Intelligence: The Search for a Catalog of Intelligence Capabilities,” in *Proceedings of the 3rd Workshop on Natural Language for Artificial Intelligence* (M. Alam, V. Basile, F. Dell’Orletta, M. Nissim, and N. Novielli, eds.), vol. 2521, pp. 1–15, Rende, Italy. CEUR-WS, 2019.
- [12] D. Kelley, “*The Art of Reasoning: An Introduction to Logic and Critical Thinking*,” W.W. Norton & Company, New York, NY, fourth edition, 2014.
- [13] S. Legg and M. Hutter, “Universal Intelligence: A Definition of Machine Intelligence,” *Minds and Machines*, vol. 17, pp. 391–444, 2007b.
- [14] D. Monett and C.W.P. Lewis, “Definitional Foundations for Intelligent Systems, Part I: Quality Criteria for Definitions of Intelligence,” in *Proceedings of The 10th Anniversary Conference of the Academic Conference Association* (J. Vopava, V. Douda, R. Kratochvíl, and M. Konečki, eds.), pp. 73–80, Prague, Czech Republic. MAC Prague Consulting Ltd., 2020.
- [15] N. Ide, “Introduction: The Handbook of Linguistic Annotation,” in *Handbook of Linguistic Annotation* (N. Ide and J. Pustejovsky, eds.), pp. 1–18. Springer, Dordrecht, 2017.
- [16] M. Neves and J. Ševa, “An extensive review of tools for manual annotation of documents,” *Briefings in Bioinformatics*, vol. 22, no. 1, pp. 146–163, 2021.
- [17] D. Monett and C.W.P. Lewis, “Getting clarity by defining Artificial Intelligence—A Survey,” in *Philosophy and Theory of Artificial Intelligence* (V.C. Müller, ed.), SAPERE vol. 44, pp. 212–214. Springer, Berlin, 2018.
- [18] S. Legg and M. Hutter, “A Collection of Definitions of Intelligence,” in *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms* (B. Goertzel and P. Wang, eds.), vol. 157, pp. 17–24. IOS Press, UK, 2007a.
- [19] R. Klinger and P. Cimiano, “The USAGE review corpus for fine grained multi lingual opinion analysis,” in *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pp. 2211–2218, Reykjavík, Iceland. European Language Resources Association, 2014.
- [20] M. Sänger, “Aspektbasierte Meinungsanalyse von Bewertungen mobiler Applikationen,” Master Thesis, Humboldt-Universität zu Berlin, 2018.
- [21] J. Cohen, “A coefficient of agreement for nominal scales,” *Educational and Psychological Measurement*, vol. 20, pp. 37–46, 1960.

- [22] J.R. Landis and G.G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, pp. 159–174, 1977.
- [23] L.S. Gottfredson, "Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography," *Intelligence*, vol. 24, pp. 13–23, 1997.
- [24] R.J. Haier, "*The Neuroscience of Intelligence*," Cambridge University Press, New York, NY, 2017.
- [25] D. Monett, "*Examples of annotated definitions of intelligence*," The AGI Sentinel Initiative, AGISI.org, 2021. Retrieved from <https://bit.ly/AnnotatedDefsIntelligence> (Last accessed: the last date you accessed the collection).
- [26] A. Sloman, "*The Computer Revolution In Philosophy: Philosophy, science and models of mind*," Harvester Press, Sussex, revised, online edition, 2019.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/367510236>

# Exploiting Simu5G for generating datasets for training and testing AI models for 5G/6G network applications

Article · January 2023

DOI: [10.1016/j.softx.2023.101320](https://doi.org/10.1016/j.softx.2023.101320)

CITATIONS

0

READS

115

4 authors, including:



Giovanni Nardini

Università di Pisa

62 PUBLICATIONS 792 CITATIONS

[SEE PROFILE](#)



Pietro Ducange

Università di Pisa

88 PUBLICATIONS 2,261 CITATIONS

[SEE PROFILE](#)



Giovanni Stea

Università di Pisa

176 PUBLICATIONS 2,358 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Project EuQoS: End-to-End Quality of Service over Heterogeneous Networks [View project](#)



Project Network Calculus [View project](#)



## Original software publication

## Exploiting Simu5G for generating datasets for training and testing AI models for 5G/6G network applications

Giovanni Nardini <sup>a,b,\*</sup>, Alessandro Noferi <sup>a</sup>, Pietro Ducange <sup>a</sup>, Giovanni Stea <sup>a</sup><sup>a</sup> Dipartimento di Ingegneria dell'Informazione, University of Pisa, Largo L. Lazzarino 1, 56122 Pisa, Italy<sup>b</sup> Center for Logistic Systems, University of Pisa, Via dei Pensieri 60, 57142, Livorno, Italy

## ARTICLE INFO

## Article history:

Received 20 September 2022

Received in revised form 16 January 2023

Accepted 19 January 2023

## Keywords:

simu5G

Artificial intelligence

Network simulation

Dataset

## ABSTRACT

Researchers working on Artificial Intelligence (AI) need suitable datasets for training and testing their models. When it comes to applications running through a mobile network, these datasets are difficult to obtain, because network operators are hardly willing to expose their network data or to open their network to experimentation. In this paper we show how Simu5G, a popular 5G network simulator based on OMNeT++, can be used to circumvent this problem: it allows users to log data at arbitrary spatial and temporal resolution, belonging to every network layer – from the application to the physical one.

© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## Code metadata

## Current code version

v1.2.1

Permanent link to code/repository used for this code version

<https://github.com/ElsevierSoftwareX/SOFTX-D-22-00294>,[https://github.com/Unipisa/Simu5G/releases/tag/dataset\\_generator\\_software-x](https://github.com/Unipisa/Simu5G/releases/tag/dataset_generator_software-x)

## Permanent link to reproducible capsule

GNU Lesser General Public License V. 3

Legal code license

git

Code versioning system used

C++, Network Description (NED) language, few python routines.

Software code languages, tools and services used

Requires OMNeT++

Compilation requirements, operating environments and dependencies

<http://simu5g.org>, <https://github.com/Unipisa/Simu5G>

If available, link to developer documentation/manual

giovanni.nardini@unipi.it

Support email for questions

## 1. Motivation and significance

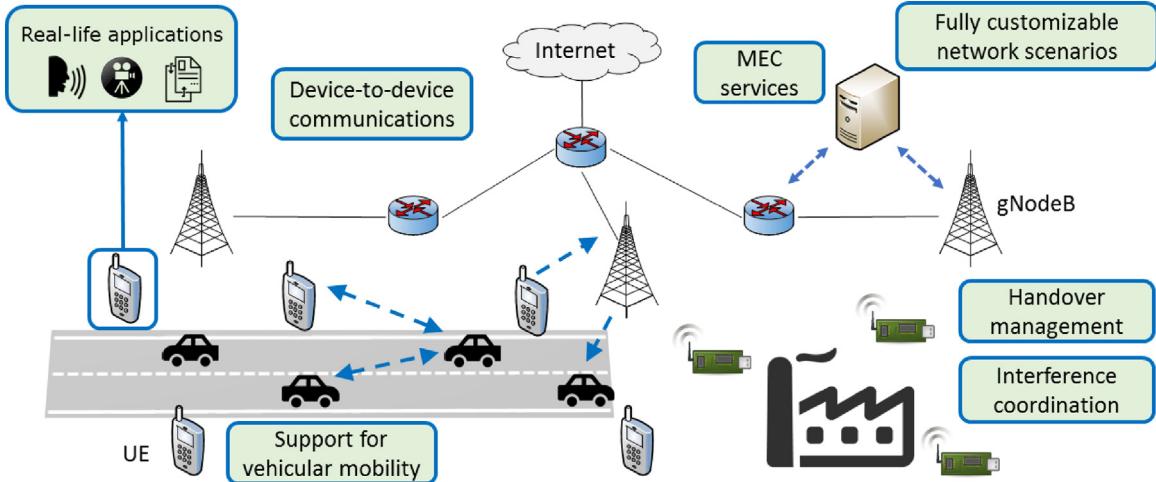
Research on Artificial Intelligence (AI) for networking has seen an enormous upsurge in the last years. AI solutions are now being envisaged to solve complex networking problems, such as optimizing the network configuration and predicting user Quality of Experience (QoE) [1]. Parallel to this, *networked* paradigms for AI algorithms, such as Federated Learning [2], are emerging: their performance (e.g., speed of convergence) depends on network Quality of Service (QoS), and networks themselves have to manage new workload for supporting such new paradigms. This double interplay – i.e., AI for networks, on one hand, and *networked* AI, on the other – has at its core cellular access (CA), i.e., 4G, 5G and Beyond-5G (B5G) [3]. Indeed, CA ensures the

expected reliability and ubiquitous diffusion, and supports complex communication/computation services via Multi-access Edge Computing (MEC) [4].

Suitable datasets for training and testing AI models for specifics applications may be hard to obtain. For example, an AI model for predicting user QoE needs to correlate data coming from different network (sub)layers and the position data on the mobile users, at the relevant time and space resolution, with the actual level of QoE perceived by users. Network infrastructure providers have their own logging procedures. However, they are (understandably) not always willing to share them, due to a plethora of reasons (e.g., competitive advantage, or the extra work required for GDPR compliance). Moreover, these data may not fit the requirements for the training algorithm: e.g., they may not log some relevant information, or log it at the wrong time/space resolution, or in suboptimal network conditions (e.g., too lightly loaded cells). The same problem occurs – possibly exacerbated – when Federated Learning of AI models is required. In this case,

\* Corresponding author at: Dipartimento di Ingegneria dell'Informazione, University of Pisa, Largo L. Lazzarino 1, 56122 Pisa, Italy.

E-mail address: [giovanni.nardini@unipi.it](mailto:giovanni.nardini@unipi.it) (Giovanni Nardini).



**Fig. 1.** Overview of Simu5G's functionalities, which cover both application- and network-level modeling of end-to-end applications, as well as flexible configuration of users mobility.

datasets should in fact include input data to all the federated entities (typically, users of cellular networks).

In this paper, we discuss how these issues can be solved by using Simu5G, a popular 4G/5G cellular network simulation library [5–7]. Simu5G is an *end-to-end* simulator for evaluating how the network affects application-layer metrics, and how applications impact network performance. It allows users to generate arbitrary network scenarios, which include user mobility, handover, real applications, and to set arbitrary probes in the code, to measure what they need at the relevant time/space granularity, with full control over experimental conditions. Simu5G can therefore be used to generate flexible datasets for training and testing AI models. We describe how to set probes in Simu5G and how to generate datasets, with reference to a real-life example for tele-operated driving. AI models learn their structure, namely their parameters, from available data. Accordingly, the higher the capability of a simulator to generate realistic data, the better the AI models will be able to act in real world applications. Simu5G's physical layer reporting has been validated according to 3GPP guidelines [5], and its MEC model has been validated in [6]. While this does not constitute any a-priori guarantee, it is however encouraging.

Simu5G is not the only software that simulates 5G networks. Authors of [8,9] discuss *physical-layer simulators* for evaluating physical-layer design (e.g., antenna performance, transmission schemes, spectral efficiency). These simulators often lack upper network protocols and cannot support real applications. Other *end-to-end* simulators include some – but not all – Simu5G's features [10,11]. For instance, 5G-LENA [10] is an ns3 library [12] that, to the best of our knowledge, lacks dual connectivity, network-controlled device-to-device communications, ETSI MEC modeling, and real-time capabilities.

## 2. Software description

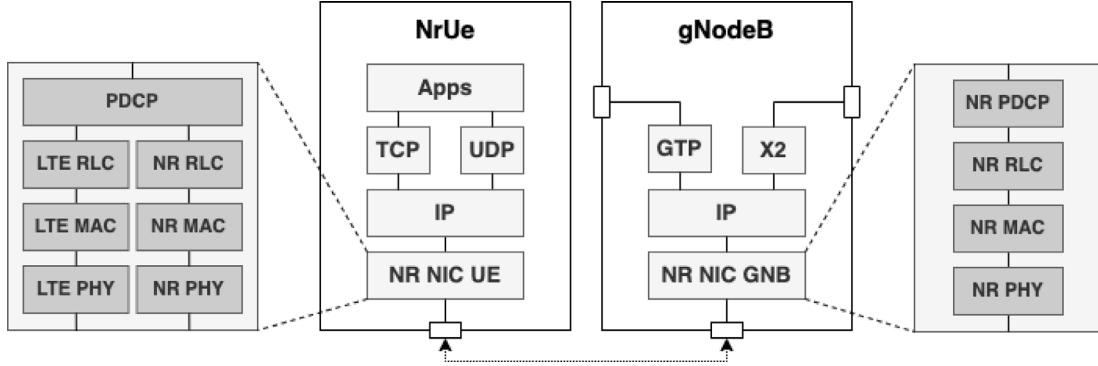
Simu5G is a discrete-event simulation library for 4G/5G New Radio networks based on OMNeT++ [13]. As shown in Fig. 1, Simu5G is interoperable with all the OMNeT++-based libraries, e.g., the INET library, which includes a wealth of TCP/IP network elements [14], libraries for vehicular mobility (e.g., using [15]), etc. Moreover, Simu5G also models the ETSI MEC standard [4, 16]. Users can therefore setup scenarios where user applications instantiate MEC applications through a 5G network, and MEC applications use the services provided by the underlying 5G network (e.g., the Radio Network Interface service). Applications can

be modeled within the simulator, or they can be external applications *interfaced* with it. Simu5G can also run in real time [7]: packets from an external application are injected into Simu5G, undergo coherent forwarding treatment (routing, delay, losses, etc.) and come out at the other end, to be conveyed to their external destination.

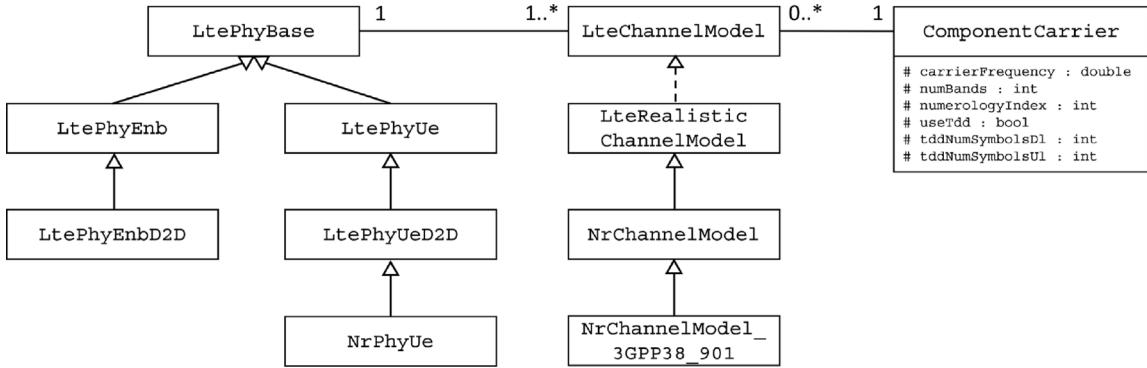
### 2.1. Software architecture

Simu5G provides a set of modules that implement the main entities of 5G New-Radio (NR) networks, such as Base Stations (BSs) and User Equipments (UEs). Following the OMNeT++ philosophy, each entity is described by a Network Description (NED) file, a declarative language that exploits inheritance and interfaces, fully convertible into XML. It defines the parameters of the module and the submodules that compose its internal architecture, as well as the gates and the connections that allows it to exchange messages with other modules. Parameters to be used in a simulation instance can be configured via an initialization (INI) file. Fig. 2 shows the architecture of the gNodeB and the NrUe modules. They both include a NR Network Interface Card (NIC) submodule, which in turn is composed of submodules representing the different layers of the NR protocol stack, from PDCP to the physical layer. The behavior of each submodule is specified by a C++ class, which defines the actions that the module performs to handle events, such as the reception of messages from another module or the expiration of a self-scheduled timer. Indeed, data-plane network packets are implemented as messages that flow through the submodules of a BS or a UE, where each layer applies its 3GPP-compliant processing before passing the messages to the downstream/upstream layer (or sending it to another entity, e.g., over the radio channel). Many NR features are also supported, such as carrier aggregation, multiple numerologies, frequency- and (flexible) time-division duplexing.

Fig. 3 reports a portion of the class diagram describing the entities performing physical-layer operations. *LtePhyBase* is the class providing the base functionalities for a node, which are specialized by specific classes at BS and UE side. Note that the same inheritance mechanism is employed for all the submodules of the NR NIC shown in Fig. 2. *LtePhyBase* references one or more channel models, i.e. one for each component carrier supported by the node. The channel model is implemented by one of the classes implementing the *LteChannelModel* interface. Each channel model is associated to one *ComponentCarrier* class, which includes the parameters of the component carrier, such as carrier frequency, numerology index, etc. An instance of *ComponentCarrier* can be used by zero or more nodes in the simulation.



**Fig. 2.** High-level architecture of Simu5G's main modules, namely the NrUe and the gNodeB modules, with special focus on the internal modeling of the NR NIC, which includes all the NR protocol layers.



**Fig. 3.** Partial view of the class diagram representing the relationship between classes at the physical layer. Inheritance is widely used to implement common functionalities and reuse as much code as possible.

## 2.2. Software functionalities

Simu5G can simulate arbitrary 5G network scenarios, as well as mixed 4G/5G and dual connectivity deployments. Scenarios can be constructed by composing modules – from Simu5G, INET and, possibly, other OMNeT++ libraries – using the NED language. The latter allows one to write parametric simulation scenarios, e.g., a multicell 5G network with a variable number of BSs and UEs. Actual simulation *parameters*, e.g., traffic generation rate of an application, number of UEs served by a BS, etc., can be configured separately in INI files, so that a set of simulation experiments can be constructed by computing the cartesian product of all the factors.

Users willing to obtain a dataset for a particular service or application will need to configure a simulation scenario where UEs run their application of interest. Simu5G comes with a set of built-in applications (i.e., application-layer modules) representing both real-life applications – such as VoIP – and generators of synthetic traffic – e.g., Constant Bit Rate. Occasionally, users may need custom applications. In this regard, the modular nature of Simu5G allows one to easily plug new application modules. This can be done by defining the NED and C++ files for that application and configuring the INI so that UEs run that application.

In addition, users need to identify the statistics of interest that will form the dataset. Simu5G already provides a large set of pre-defined metrics at both BS and UE granularity, like cell-wise/UE throughput and latency measurements at multiple levels of the network protocol stack. For instance, the end-to-end latency of transmitted packets can be obtained at both the application and the MAC level. Custom statistics can also be defined. Statistics are declared within modules, namely in the NED file describing them,

and probes capturing the samples are implemented within the corresponding C++ files. Statistics recording is based on *signals*. The latter are generated by a module and received by whichever other component of the simulation registered a *listener* for them. To record metrics, when a module computes a new sample, it emits a signal carrying it, which is then received by a *recorder* that stores and elaborates it to build complex statistics. Listings 1–4 exemplify the code that must be implemented to add a new statistic recording the end-to-end delay of packets received by a custom application module.

By default, each simulation instance produces two output files storing the statistics, namely a .sca file including all scalar-type statistics and a .vec file including all vector-type statistics. Since not all statistics may be of interest to the user in a particular scenario, the actual set of statistics that a simulation produces can be fine-tuned via the INI file. If statistic *statA* is defined in submodule *Y* of module *X*, a user can disable all scalars related to it by configuring:

\*\*.<moduleX>.<submoduleY>.statA.scalar-recording=false.

Likewise, the *vector-recording* parameter can be used to control vector statistics. The hierarchical structure of the simulation scenario allows one to disable all statistics generated by a module or by the whole simulation – by using wildcards.

Output files usually need to be parsed to generate the final dataset. Being raw text files, they can be parsed using any kind of scripting. However, OMNeT++ includes tools that facilitate data extraction, e.g. an analysis tool that allows the user to browse statistics, filter them (e.g., by name, module, replicas, type) and create plots. Moreover, it exports selected statistics to JSON or CSV files (e.g., for further analysis through spreadsheet applications), or other formats readable (e.g.) by R and Matlab.

*Listing 1: Definition of the signal in the NED file*

```
@signal[endToEndDelaySignal](type=simtime_t);
@statistic[endToEndDelay](title="End-to-end delay"; source="endToEndDelaySignal";
record=mean, vector);
```

*Listing 2: Declaration of the signal in the C++ class header file*

```
omnetpp::simsignal_t endToEndDelaySignal_;
```

*Listing 3: Registration of the signal at the simulation initialization in the C++ class implementation*

```
endToEndDelaySignal_ = registerSignal("endToEndDelaySignal");
```

*Listing 4: Emitting one sample value of the signal in the C++ class implementation*

```
simtime_t delay_val = rxTime - txTime;
emit(endToEndDelaySignal_, delayVal);
```

Alternatively (e.g., if the user does not have access to a GUI), the *opp\_scavetool* program included with OMNeT++ allows one to extract statistics from the command line.

### 3. Illustrative examples

In this example, we consider the use case of Tele-operated Driving (ToD), where vehicles can be remotely driven by a human operator. With ToD, the operator must receive a high-definition, real-time video streaming from the vehicle through the mobile network, so that he/she can be aware of the environment around the vehicle and control the latter safely. In this scenario, QoE plays a crucial role, as the ToD functionality could not be exploited if the video presents impairments at the receiving side (i.e., the remote driver). In this context, AI models may be trained for predicting possible QoE degradations in the near future, hence allowing the remote driver to take proper countermeasures, such as parking the vehicle in a safe location or requesting the passenger to take control of the vehicle. The Simu5G software can be exploited to simulate the above scenario and generate a dataset that will be used by AI experts to properly design the parameters of specific AI models. Unless specified otherwise, in the following we refer to files provided within folder *simulations/NR/videostreaming\_dataset\_generator*.

With reference to Fig. 4, the NED file defines a simulated network including three gNBs, namely *gnb1*, *gnb2* and *gnb3*. The latter provide the radio coverage over a main road intersected by three secondary roads. Traffic lights regulate the traffic at crossroads. Five UEs are deployed in the floorplan, each of them sending a real-time video stream to their corresponding receiving application residing at the MEC host. Eight *background* gNBs (i.e., gNBs whose only purpose is to produce realistic interference on *foreground* gNBs) complete the picture.

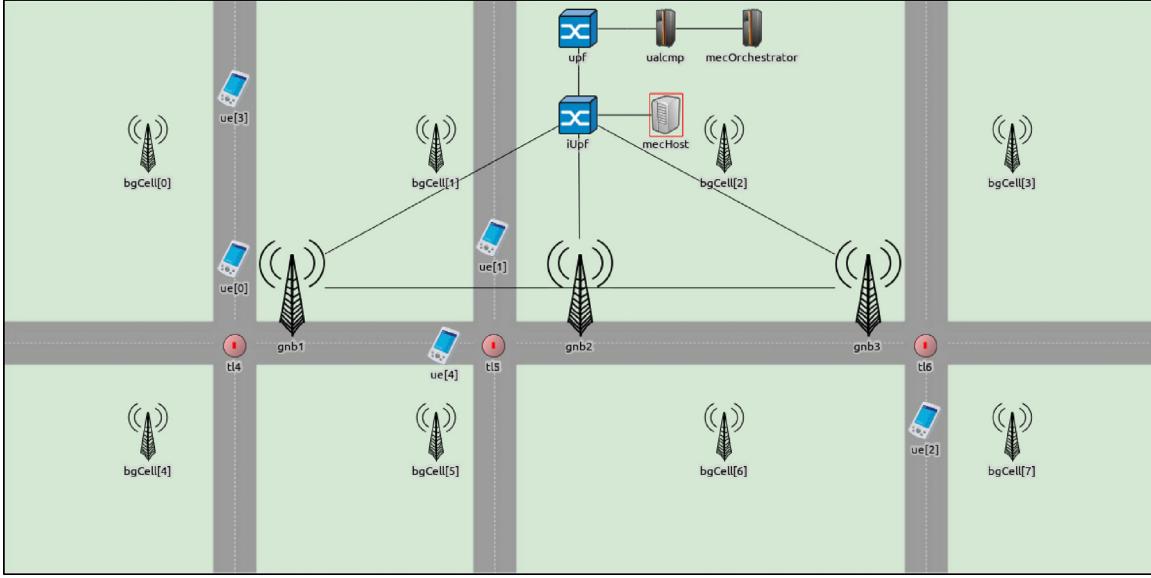
The above scenario is parametrized by the *omnetpp.ini* file, which is organized in several sections identified by the label in square brackets. The *General* section specifies the NED file containing the network and network-wise 5G parameters, such as carrier aggregation, handover, transmit powers and MEC system configurations. Moreover, it specifies the simulation duration (set to 900s) and the number of scenario repetitions (set to 10): the OMNeT++ environment will automatically generate ten independent replicas, each 15-minutes long and using different

seeds for the pseudo-random generator. The *UrbanNetwork* section, instead, defines the parameters specific to the simulated network. For example, it configures the position of both foreground and background gNBs, as well as the number of UEs (set to five) and their mobility type. In this scenario, UEs use a custom mobility module called *TrafficLightMobility*, (see folder *src/mobility/trafficLightMobility*) which makes UEs move in a straight line at constant speed, randomly turn at crossroads, and stop when they encounter a red traffic light. This section of the INI file also defines the application module that is run by the UEs.

For the use-case presented in this paper, we implemented a custom module that models the behavior of a real-time video-streaming application at both sender and receiver side, the latter being implemented as a MEC app. The implementation can be found in folder *src/apps/RealTimeVideoStreamingApp*. The sender application generates video frames according to a trace file, each line of which specifies the size of a video frame. In our simulations, trace files were obtained from dash-camera videos using the *FFmpeg* library. For each frame, the sender generates a number of fragments and sends them via the UDP protocol. At the receiving side, the application tries to reconstruct the complete frame and play them out at the intended time. The relevant statistics for the QoE prediction – such as the inter-arrival time, frame and fragment size, percentage of frame displayed, and so on – were declared in the NED file of the sender application, i.e. *RTVideoStreamingSender.ned*. In order to generate a dataset representing the values of such metrics over time, statistics are declared as *vector* in the NED file.

In the C++ class, the corresponding signal were declared in the *.h* file, and registered (through the *registerSignal()* function) in the *handleUeMessage()* function in the *.cc* file. Then, statistics are emitted in the proper function handling the reception of fragments and the playout of frames. For example, the *playoutFrame()* is invoked when a frame has to be played out, it retrieves the number of received segment belonging to that frame and computes the percentage of the frame correctly received by dividing the total size of the received segments by the total size of the frame. The resulting value is emitted as a sample of the *frameDisplayed* statistic by invoking *ueAppModule->emit(frameDisplayed\_, percentage)*.

With reference to Fig. 5, the *omnetpp.ini* can be configured to handle which statistics are produced and where they are saved.



**Fig. 4.** Simulated scenario: 5G-enabled vehicles move along the roads deployed in the floorplan while sending a video stream using the Real-time Transport Protocol (RTP) over the radio network to the MEC Host, which is connected to the 5G core network.

```
#####
Statistics #####
**.rtVideoStreaming*.vector-recording = true
**.avgServedBlocksUl*.vector-recording = true
**.averageCqIUl*.vector-recording = true
**.rcvdSinrUl*.vector-recording = true
**.measuredSinrUl*.vector-recording = true
**.servingCell*.vector-recording = true

**.scalar-recording = false
**.vector-recording = false

output-scalar-file = ${resultdir}/${configname}/${iterationvars}-${repetition}.sca
output-vector-file = ${resultdir}/${configname}/${iterationvars}-${repetition}.vec
```

Selection of relevant metrics to be recorded.  
All other metrics are not recorded to save disk space and speed-up the simulations

Path where statistics are saved at the end of the simulation campaign. OMNeT++ variables are used to construct the path or file name

**Fig. 5.** Statistics filtering in the omnetpp.ini file: the blue box at the top includes the settings required to filter which metrics are recorded by the simulation, whereas the orange box at the bottom includes parameters that specify the output path where statistics can be retrieved at the end of the simulation.

Once the scenario is setup, we can launch the simulation campaign by typing `opp_runall simu5 g -u Cmdenv -c UrbanNetwork` from the command line. Raw statistics are saved in the files specified by the `output-vector-file` parameter in `omnetpp.ini` file, as shown in Fig. 5. In order to obtain usable data, we must parse the above files, extract the relevant statistics and produce a dataset with the desired format. Parsing and extraction of the statistics is done via the `opp_scavetool` program. This produces an intermediate CSV file, which can be parsed further to generate the final dataset. To do so, we implemented a python script, namely `DatasetExtractor.py` that filters unnecessary columns and cleans some text in order to reduce the size of the generated dataset.

The final dataset is a set of tuples with the following format:  
`<run, module, metric, timestamps, values>`  
Where:

- `run` is the unique identifier of the simulation that generated the metric in the tuple;
- `module` is the entity that produced the metric, e.g., `ue[0]`;
- `metric` is the name of the collected metric;
- `timestamps` and `values` are two arrays. Their  $i$ th elements,  $t_i$  and  $m_i$  concur to represent point  $(t_i, \text{metric}(t_i))$ .

The final rough dataset may be re-elaborated for creating a typical dataset for training and testing AI model. Usually, the AI-dataset is a collection of input-output tuples. Each tuple includes a target value to be predicted and a list of input values. Thus, specific input variable and the output variable must be extracted from the rough dataset.

#### 4. Impact

Despite its code having been publicly available for less than 18 months, Simu5G is already being used by a large community of researchers: it has been downloaded more than 4100 times so far, from all over the world – see Fig. 6. Among the 45+ papers already citing [5],<sup>1</sup> 12 are from (other) research groups that use it to validate their research, some of which involve AI techniques [17–20].

Simu5G has been developed in the framework of a joint research project between the University of Pisa and Intel. It is currently being used within the Hexa-X project, EU's 6G flagship project [3], where it supports research, development and demonstration activities for Federated Learning of Explainable AI (XAI) models for QoE prediction. Recently, a preliminary performance

<sup>1</sup> Google Scholar search, Sept. 6, 2022



**Fig. 6.** Google analytics report of the Simu5G website [1] from the release of the code (Apr 15, 2021) to Sep 6, 2022. The rightmost graph shows the number of downloads per country.

analysis was presented in [1], where rough data generated using Simu5G were elaborated to create a dataset for estimating QoE values using statistics extracted from network metrics. The dataset is publicly available at [21].

Simu5G is also listed among the tools for MEC app developers in the ETSI MEC ecosystem [22] – in fact, MEC app developers can use it as a cradle to test their production-level code for performance, in realistic and customizable 5G scenarios.

## 5. Conclusions

This paper discussed how datasets for AI research on 4G/5G/5G networking can be generated by using Simu5G, a popular open-source simulation library. In fact, Simu5G – that simulates all the networking stack, from the application to the physical layer, includes application-level features like, e.g., MEC hosting, and supports integration with OMNeT++-based libraries like, e.g., Veins for vehicular mobility [23] – allows users to record metrics of heterogeneous provenance, that can be used to train and test machine-learning algorithms. We have shown how a user can add arbitrary probes in the code, to record any metrics she needs in her dataset, with the desired time/space resolution.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The code and the resulting data have been published in the Simu5G GitHub repository

## Acknowledgments

Work partially supported by the Italian Ministry of Education and Research, Italy in the framework of the CrossLab project (Departments of Excellence), and by the European Commission through the H2020 projects Hexa-X (Grant Agreement no. 101015956). We thank our colleagues Francesco Marcelloni and Alessandro Renda for their useful discussions on this paper.

## References

- [1] Corcuera Bárcena JL, Ducange P, Marcelloni F, Nardini G, Noferi A, Renda A, et al. Towards trustworthy AI for QoE prediction in 5G/6G networks. In: First international workshop on artificial intelligence in beyond 5G and 6G wireless networks. Padua, IT; 2022, p. 18–23.
- [2] Renda A, et al. Federated learning of explainable AI models in 6G systems: Towards secure and automated vehicle networking. Information 2022;13(8):395. <http://dx.doi.org/10.3390/info13080395>.
- [3] Hexa-X project. 2023, website: <https://hexa-x.eu> [Accessed January 2023].
- [4] ETSI white paper (28) MEC in 5G networks. 2018, link: <https://bit.ly/3bzCLFI> [Accessed January 2023].
- [5] Nardini G, Sabella D, Stea G, Thakkar P, Virdis A. Simu5G—an OMNeT++ library for end-to-end performance evaluation of 5G networks. IEEE Access 2020;8:181176–91. <http://dx.doi.org/10.1109/ACCESS.2020.3028550>.
- [6] Noferi A, Nardini G, Stea G, Virdis A. Rapid prototyping and performance evaluation of ETSI MEC-based applications. Elsevier Simul Model Pract Theory 2023;123:102700. <http://dx.doi.org/10.1016/j.smpat.2022.102700>.
- [7] Nardini G, Stea G, Virdis A. Scalable real-time emulation of 5G networks with Simu5G. IEEE Access 2021;9:148504–20. <http://dx.doi.org/10.1109/ACCESS.2021.3123873>.
- [8] Kim Y, et al. 5G-K-simulator: 5G system simulator for performance evaluation. In: 2018 IEEE international symposium on dynamic spectrum access networks. Seoul, Korea (South; 2018, p. 1–2. <http://dx.doi.org/10.1109/DySPAN.2018.8610404>.
- [9] Müller M, Ademaj F, Dittrich T, et al. Flexible multi-node simulation of cellular mobile communications: The vienna 5G system level simulator. J Wireless Com Netw 2018;2018:227. <http://dx.doi.org/10.1186/s13638-018-1238-7>.
- [10] Patriciello N, Lagen S, Bojovic B, Giupponi L. An E2E simulator for 5G NR networks. Simul Model Pract Theory 2019;96:101933. <http://dx.doi.org/10.1016/j.smpat.2019.101933>.
- [11] Martiradonna S, Grassi A, Piro G, Boggia G. 5G-air-simulator: An open-source tool modeling the 5G air interface. Comput Netw 2020;173(22):107151. <http://dx.doi.org/10.1016/j.comnet.2020.107151>.
- [12] Ns3 network simulator. 2023, Website: <https://www.nsnam.org> [Accessed January 2023].
- [13] OMNeT++ discrete event simulator. 2023, Website: <https://omnetpp.org> [Accessed January 2023].
- [14] INET framework. 2023, Website: <https://inet.omnetpp.org> [Accessed January 2023].
- [15] Sommer C, German R, Dressler F. Bidirectionally coupled network and road traffic simulation for improved IVC analysis. IEEE Trans Mob Comput 2011;10(1):3–15. <http://dx.doi.org/10.1109/TMC.2010.133>.
- [16] ETSI GS MEC 013 v2.2.1. In: Multi-access edge computing. Location API; 2022.
- [17] Batista JOR, da Silva DC, Martucci M, Silveira RM, Cugnasca CE. A multi-provider end-to-end dynamic orchestration architecture approach for 5G and future communication systems. Appl Sci 2021;11(24):11914. <http://dx.doi.org/10.3390/app112411914>.
- [18] Nguyen AC, Pamuklu T, Syed A, Kennedy WS, Erol-Kantarci M. Reinforcement learning-based deadline and battery-aware offloading in smart farm IoT-UAV networks. In: ICC 2022 - IEEE international conference on communications. Seoul, Korea, Republic of; 2022, p. 189–94. <http://dx.doi.org/10.1109/ICC45855.2022.9838500>.

- [19] Antonio G-P, Maria-Dolores C. AIM5la: A latency-aware deep reinforcement learning-based autonomous intersection management system for 5G communication networks. Sensors 2022;22(6):2217. <http://dx.doi.org/10.3390/s22062217>.
- [20] Rehman A, Haseeb K, Saba T, Lloret J, Sendra S. An optimization model with network edges for multimedia sensors using artificial intelligence of things. Sensors 2021;21(21):7103. <http://dx.doi.org/10.3390/s21217103>.
- [21] Dataset for QoE prediction in B5G/6G networks. 2023, [Accessed January 2023].
- [22] MEC ecosystem wiki. 2022, [Accessed May 2022].
- [23] Nardini G, Virdis A, Stea G. Simulating cellular communications in vehicular networks: Making SimuLTE interoperable with Veins. In: Proc. of the 4th OMNeT++ community summit. Bremen, DE; 2017, p. 7–8. <http://dx.doi.org/10.48550/arXiv.1709.02208>.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/365873592>

# Human Behavior Models for Adaptive Training in Mixed Human-Agent Training Environments

Conference Paper · November 2022

---

CITATIONS  
0

READS  
29

1 author:



Some of the authors of this publication are also working on these related projects:

-  Smart Bandits - Machine Learning [View project](#)
  
-  CIGA - Connecting Agents to Virtual Environment [View project](#)

# Human Behavior Models for Adaptive Training in Mixed Human-Agent Training Environments

Joost van Oijen  
NLR – Royal Netherlands Aerospace Centre  
Amsterdam, the Netherlands  
Joost.van.Oijen@nlr.nl

## ABSTRACT

Current trends in simulation-based military training show an increasing demand for artificial intelligence (AI) and data science technologies to offer more flexible and adaptive training solutions for military personnel in rich simulated training environments. This requires technologies capable of (1) measuring and assessing performance of trainees in real-time, and (2) simulating role-playing agents that can replace human role-players and adapt their behavior to guide learning experiences for a trainee.

The above requirements pose two challenges. First, current training systems often lack an understanding of the dynamic context of a trainee's behavior, which is required to judge its performance during training of specific missions, (part-)tasks or tactical situations. This context is typically only available in the head of an observing instructor. Second, behavior models for simulated role-players are often black boxes from the point of view of an instructional system and cannot easily be adapted for instructional purposes (e.g. exhibiting degraded performance or making deliberate mistakes).

In this paper we present a unified human behavior modeling (HBM) approach that addresses the above challenges. It is based on the idea that HBMs can be used to model roles for both human trainees and agents. For an agent, it acts as an AI model that produces behavior, equipped with predetermined adaptive variables. For a trainee, it acts as an observer that tracks and measures behavior being performed by that trainee. As a computational HBM approach, we examine the use of the Context-based Reasoning (CxBR) modeling paradigm, allowing context-specific behavior and performance modeling. The HBMs (1) support instructor-based and automated tutoring, (2) promote collaborative design of instructional systems between training designers, subject-matter experts and behavior modelers, and (3) allow for interchangeable roles between trainees and agents in training. We demonstrate a proof-of-concept of the HBM approach in the scope of a training system for military aircrew training.

## ABOUT THE AUTHORS

**Joost van Oijen, PhD** is a Senior Scientist at the Royal Netherlands Aerospace Centre (NLR). Having a background in Computer Science and Artificial Intelligence, he has over ten years of experience in AI for modeling & simulation, both in the industry and academia. At his current position, Joost leads several R&D projects focused on human behavior modeling for training and decision support. Having a strong software engineering background, he is actively involved in the development of multi-agent systems and behavior modeling tools for military simulation systems.

# Human Behavior Models for Adaptive Training in Mixed-Agent Training Environments

Joost van Oijen

NLR – Royal Netherlands Aerospace Centre

Amsterdam, the Netherlands

[Joost.van.Oijen@nlr.nl](mailto:Joost.van.Oijen@nlr.nl)

## INTRODUCTION

Current trends in simulation-based military training show an increasing demand for artificial intelligence (AI) and data science technologies to offer more flexible and adaptive training solutions for military personnel in rich simulated training environments. Such technologies contribute to adaptive training in several ways:

- First, one can benefit from the use of simulated AI role-players (hereafter named agent role-players) that are capable of taking on the role of adversaries or team-members. In the context of military simulation, these are also known as Computer Generated Forces (CGF). The use of simulated role-players has the benefit of being able to replace human role-players, allowing for more flexible training schedules, requiring fewer personnel, and thus saving costs.
- Second, data analytics can be used to measure and obtain real-time objective data on trainee performance in the simulation, in regard to tasks or skills to be trained. Currently instructors often rely on subjective self-observations in order to assess trainee performance, either real-time during training or during after-action-review based on replays.
- Finally, there is a desire to tailor training scenarios to the skill level of the individual trainee, in line with upcoming training methodologies such as performance-based and adaptive training. This includes the ability to adapt the behavior of agent role-players for instructional purposes in order to guide the learning experiences for a trainee. Currently, agent role-player behavior models are often ‘black boxes’ from the point of view of an instructional system and are not always designed to be externally adapted.

In this paper we present a unified human behavior modeling (HBM) approach that combines the above capabilities in a synergetic modeling approach for adaptive training systems. The approach is based on the idea that HBMs can be used to model roles for both human trainees and agent role-players. For an agent role-player, it acts as a behavior model that *produces* behavior, equipped with predetermined adaptive variables for instructional control. For a human trainee, it acts as an *observer* that tracks and measures behavior being performed by a trainee in the simulation. The proposed approach (1) supports instructor-based and automated tutoring, (2) promotes collaborative design of instructional systems between training designers, subject-matter experts and behavior modelers, and (3) allows for interchangeable roles between human trainees and agent role-players in training.

In the remainder of this paper we start by sketching a background based on related work, followed by our contributions in subsequent sections:

1. We describe the theory of the HBM approach and its relation with instructional systems.
2. We examine a computational approach using the context-based reasoning (CxBR) paradigm.
3. We show how the approach can be used in conjunction with training design by using a case study in the domain of military air combat training.
4. We present a technical proof-of-concept for integrating HBMs in an existing training system.

This paper describes the first phase of a study towards the implementation of the HBM approach for adaptive training, focusing on the theoretical and technical framework. In the next phase of the study we will apply the framework using expert-validated adaptive training scenarios in the air combat domain, from a more sound instructional design point of view.

## **BACKGROUND**

The work presented in this paper touches upon three areas of related research. The first area is concerned with behavior modeling techniques for CGFs. The second area is concerned with techniques for measuring and analyzing performance of human trainees in simulations. The third area relates to techniques for using adaptive agents in systems known as Adaptive Instructional Systems (AIS). Below these areas are shortly described, followed by a reflection on the relation to our work.

### **Computer Generated Forces**

Computer Generated Forces (CGF) are autonomous military actors that are used in military simulation for training or decision-support applications (Abdellaoui et al., 2009). There is a wide range of different behavior modeling techniques and paradigms that have been used to model CGF behaviors. Techniques range from more practical control-based techniques such as scripts, state machine or behavior trees; to more behavior-oriented approaches based on human notions such as goals, beliefs and plans (e.g. the belief-desire-intentions (BDI) paradigm); to approaches that aim to simulate human cognitive processes, based on cognitive models and architectures (e.g. ACT-R, SOAR). Within NATO, several studies have been performed on human behavior representation and models for simulation, both from an engineering perspective (Lewis, 2019) and human factors perspective (Lotens et al., 2009). Specific to the air combat domain, behavior modeling for fighter pilots in simulation-based training is a well-researched topic (Doyle & Portrey, 2014), (Dong et al., 2019).

### **Performance Measurement in Simulation**

The increase of simulation-based training facilitates the implementation of training methodologies such as performance-based or adaptive training where instruction can be tailored to the performance of an individual trainee. Human-in-the-loop simulations provide a controlled training environment where data can be gathered in real-time on the task performance of humans in a simulation, as well as on psychophysiological states they experience. Based on collected data, performance analytics can give insight into a trainee's technical and non-technical skills during training, and serve as a motivation for adaptations of the training environment. There exists many theories and methodologies for performance measurement in simulation-based training (Salas et al., 2009). Specific to the air combat domain, numerous studies have addressed performance modeling for both technical and non-technical skills and competencies of fighter pilots. For instance, (Arar & Ayan, 2013) proposed a framework for automated measurement of task performance, based on air combat performance metrics obtained from a simulation system. Tools such as PETST<sup>TM</sup> (Portrey et al., 2006) have been developed for collecting performance metrics from simulations to support pilot assessment research (Rowe et al., 2008) (Freeman et al., 2020). In (Mansikka et al., 2021), a performance model is introduced that integrates measures such as situation awareness and mental workload, in addition to objective task performance measures. In other studies, sensors external to the simulation have been employed to measure mental states such as cognitive workload using EEG (Mohanavelu et al., 2020), or engagement using data from head-mounted displays (Bell et al., 2021). The latter demonstrated a policy for adaptive instruction that can be used by instructors to restore detected lapsed engagement.

### **Adaptive Agents in Adaptive Instructional Systems**

An Adaptive Instructional System (AIS) is “a computer-based system that guides learning experiences by tailoring instruction and recommendations based on the goals, needs, and preferences of each learner in the context of domain learning objectives.” (Sottilare & Brawner, 2018). The use of adaptive agents as role-players in an AIS has been addressed in recent studies. (Freeman et al., 2019) describes a testbed for developing and evaluating adaptive agents for pilot training. A set of functions are identified for agents to enable them to be employed effectively in an AIS, related to tactical proficiency and instructional efficacy capabilities. In (van den Bosch et al., 2020), a framework is presented for an AIS that uses adaptive agents whose adaptations are controlled by a director agent. (Bell & Sottilare, 2019) examine the use of agent-based services to fulfill different functions within an AIS, one of which addresses adaptive role-players as constructive behavioral services.

## Concluding

In relation to our work, the presented HBM approach does not conflict with specific technologies addressed in the above research. As a design construct, it is independent of a particular implementation for adaptive CGF behaviors or performance measurement models in a simulation. Rather, it focuses on unifying the HBM design of individual role actors to support instructional systems in (1) modeling task behaviors for agent role players, (2) modeling performance measurements for trainees and (3) integrating adaptive behavior variables for instructional purposes. We examine an approach to integrate these capabilities in a computational HBM using the Context-based Reasoning (CxBR) modeling paradigm (Stensrud et al., 2004). This paradigm allows breaking down complex behaviors into smaller units of behavior through compositional and hierarchical design, such that behaviors and performance measurements can be addressed in a context-specific manner, relating to specific training tasks, skills or competencies for a particular role. When combining these capabilities in a single model, the same model can be used interchangeably for driving role-player behaviors or measuring trainee performance.

## A UNIFIED HUMAN BEHAVIOR MODELING APPROACH

In this section we describe the rationale of a unified human behavior modeling (HBM) approach to facilitate adaptive training in a mixed human-agent environment. To explain the role of an HBM for an adaptive training system, the illustration in Figure 1 is used. Below, we first sketch the adaptive training system. Afterwards we describe the HBM approach and motivate its role.

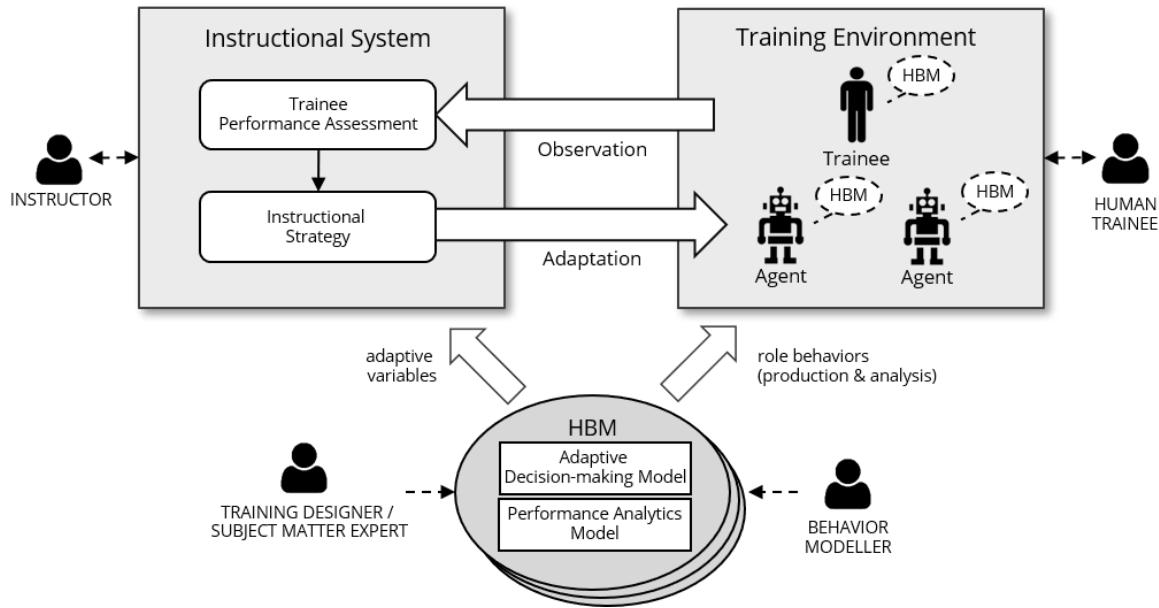


Figure 1: HBM Role in an Adaptive Training System

### Adaptive Training System

The top half of Figure 1 sketches a basic adaptive training system, consisting of a Training Environment (on the right) and an Instructional System (on the left).

The training environment represents a (human-in-the-loop) simulation environment that can be populated by intelligent actors. These actors can represent human trainees or agent role-players. The agent role-players are capable of expressing task behavior in line with their assigned role in the environment (e.g. different team-member roles). They are driven by computational behavior (AI) models capable of replacing the role of human role-players in training.

A human trainee controls the behavior of a trainee actor that represents its virtual embodiment in the simulation (e.g. through human machine interfaces offered by a simulator or virtual reality application).

The instructional system represents a computer-based training system that can observe and control the training environment in order to provide (adaptive) learning experiences for a trainee, in line with certain training objectives such as being trained on certain tasks, skills or competencies. The instructional system interfaces with the training environment by (1) being able to observe the behavior of trainees in order evaluate their performance, and (2) being able to adapt the training environment for instructional purposes (e.g. adjust the complexity of a training session). Such adaptation can relate to changing the state of physical elements of the environment, as well as changing the behavior of role-players (i.e. changing the cognitive state of agents).

The function of the instructional system is summarized by two processes shown in Figure 1, namely *Trainee Performance Assessment* and, based on the outcome of that assessment, the implementation of an *Instructional Strategy*, whose outcome orchestrates adaptations. The instructional system can be either be fully automated (also known as an Adaptive Instructional System), or controlled by a human instructor through an interface like an operating station (IOS) or dashboard.

### **Human Behavior Models**

The bottom half of Figure 1 shows a set of HBMs where each HBM can be associated with an intelligent actor in the training environment (i.e. human trainee or agent role-player). The purpose of an HBM is to provide a single, unified computational model that combines two capabilities required for the implementation of the described training system:

1. **Adaptive Decision Making model:** An HBM includes a decision-making model that can be used to drive the behavior of an agent role-player in the training environment. Additionally, explicit adaptive variables can be defined that are used to control or influence the decision-making process during training. These variables can be exposed to an instructional system that can use them to initiate available adaptations.
2. **Performance Analytics model:** An HBM includes a performance analytics model that can be used to measure behavior performance of an actor role through quantifiable metrics related to performance indicators. For an instructional system, it provides real-time observation of trainees' performance measurements.

An HBM for an agent role-player can be said to play an *active* role in the simulation, as it uses its decision-making model to *produce* behavior. An HBM for a human trainee plays a *passive* role, as it is used to *observe* behavior using the performance analytics model. As a special case, another type of actor can be considered, namely an agent learner. An agent learner uses a decision-making model with machine learning capabilities. For instance, a reinforcement learning (RL) algorithm is employed to learn certain behaviors or tasks in the environment (a policy in RL terms). In this case, the agent can be considered as the ‘trainee’ in the context of an instructional system, requiring learning guidance through feedback signals (rewards in RL terms). The use of agent learners will not be addressed further in this paper. The concept of using human-inspired adaptive instructional systems (AIS) for RL agents has been explored in an earlier study (van Oijen et al., 2021).

The key motivations for combining the above capabilities into a single HBM design construct are as follows:

#### **Capturing the cognitive dimension of the environment**

The collective of all HBM-based actors in the environment can be seen to represent the *cognitive environment*, encompassing all intelligible actors in the environment. It exists together with the *physical environment* consisting of all physical elements such as terrain, infrastructure, weather, equipment and the physical embodiments of the actors. Many current instructional systems can only observe and adapt the simulation environment at the physical level: for instance, observing the physical state of a trainee or adapting the physical state of environmental elements. The additional cognitive level offers instructional systems a more ‘intelligent’ level of observation and adaptation. For observation, it can observe the active behavior of a trainee and potentially the (inferred) intention behind that behavior, associated with metrics on the behavior’s performance. Consequently for control, it can adapt and influence the internal decision-making of agent role-players, allowing more intelligent adaptations in support of training.

**Intelligent adaptivity by design**

In the HBM, the extent of the level of adaptivity for an agent role-player is explicitly grounded in the behavior model through explicit adaptive variables. From an instructional design point of view, these variables can be seen as the ‘control dials’ for the system to adapt agent role-player behavior. The requirements for adaptive variables need to be taken into account at the start of the development process of an agent’s decision-making model, as it can be hard to add adaptivity at a later stage when the models may have already turned into ‘black-box’ solutions. When agents are not able to support the desired adaptivity, the alternative approach (which is commonly used) is to perform a (temporal) manual take-over of an agent actor in the environment. However, as agents become temporarily ‘disconnected’, this can lead to inconsistencies within the agent’s decision-making model when control of execution is returned back to the agent.

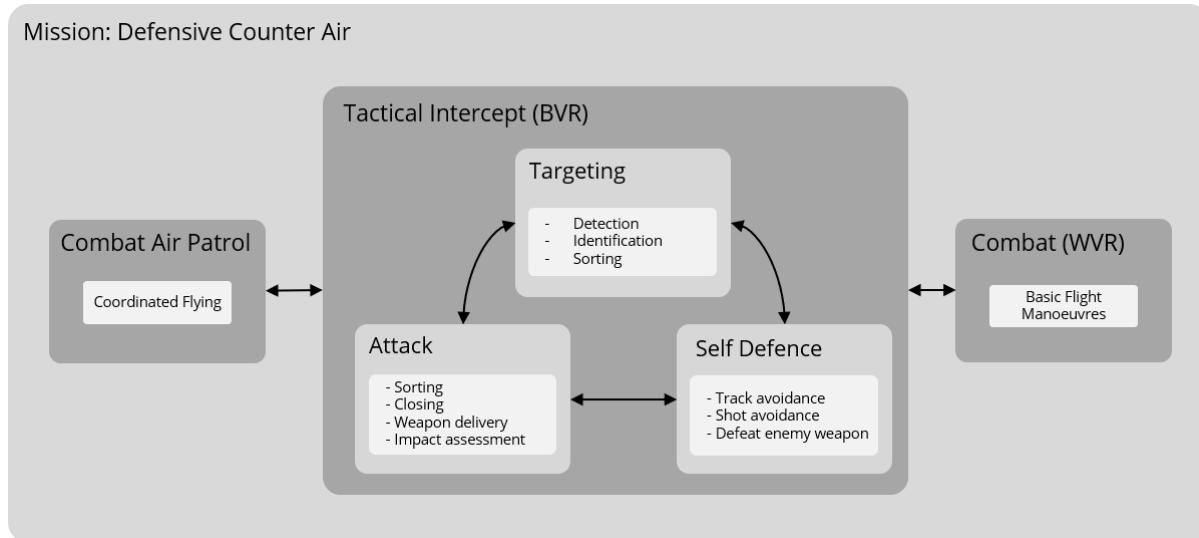
**Promotes collaborative design**

The design of an adaptive training system as sketched above is generally a collaborative effort between instructional designers, behavior modelers and subject matter experts (SME). Behavior models cannot easily be designed fully independently from instructional design. Designers and SMEs need to collaborate on (1) defining appropriate task models for actors; (2) on translating task performance indicators to measurable and quantifiable metrics in the simulation; and (3) on translating instructional needs for adaptivity to adaptive variables for role-players. These design aspects are addressed in more detail later in this paper. As the HBM combines these design efforts, this makes efficient use of required SME resources and makes it easier to support interchangeable roles between agent and human trainee actors.

**APPLYING THE CONTEXT-BASED REASONING PARADIGM**

A computational approach was investigated for the implementation of the proposed HBM design. The approach is based on the Context-based Reasoning (CxBR) paradigm (Stensrud et al., 2004). CxBR is an agent modeling paradigm for intelligent agents that focuses on tactical-oriented behaviors. It was originally proposed for modeling agents in environments where tactical expertise is required. CxBR is based on the idea that humans at any time only use a fraction of their knowledge, and that situational awareness (SA) and decision-making processes are highly driven by situational contexts.

Behavior models in CxBR are defined through a collection of contexts and transitions between those contexts. Within an active context, specific knowledge requirements and decision-making rules apply, tailored to that specific context. In CxBR, a top-level context is said to represent a *mission*. Consecutively, contexts can represent e.g. mission phases, tactical situations or (part-)tasks, which, if needed, can be decomposed further into sub-contexts. As an example, Figure 2 illustrates an example context topology for a CxBR model for an actor performing a mission in the air combat domain. It shows a decomposition of tasks related to achieving a particular mission.

**Figure 2: Example Task Model as a Context-based HBM.**

The use of contexts provides a structured analysis and common language for a task domain and can easily be understood by designers and SMEs. E.g. in related work, CxBR has been employed in a graphical design tool to facilitate the acquisition of knowledge for military tactics (Castro et al., 2002). An attractive property of the CxBR paradigm is that it does not enforce any specific technology for the implementation of behaviors within a context. Behaviors can be implemented e.g. using scripts, finite state machines, cognitive models or machine learning approaches. Additionally, contexts can be reused across CxBR models.

In this study the CxBR paradigm is used to implement the two capabilities of the HBM approach that were presented in the previous section: providing (1) an adaptive decision-making model, and (2) a performance analytics model:

### **CxBR for Adaptive Agent Role-players**

In its original design, the CxBR approach specifies the inclusion of context-specific decision-making models for agents. Additionally, it provides a built-in mechanism that can be used to define adaptive variables. In CxBR, these are denoted as so-called *moderators* that can be defined for contexts or their transitions. Moderators have been proposed to model aspects such as human mood and emotions that can affect decision-making rules and alter behaviors in a particular context (Stensrud et al., 2002). It is the responsibility of the behavior modeler to reflect the effect of moderator changes within the decision-making model. The moderators can be exposed to an external instructional system for (real-time) control in order to achieve an instructional intervention.

### **CxBR for Trainee Performance Measurement**

Besides the original purpose of the CxBR paradigm to model agent behavior, we propose to employ the paradigm also for the measurement of behavior performance. From a training point of view, instructors would typically also employ some form of context-based reasoning when assessing the performance of a trainee, in the context of a specific mission phase, procedural task, or tactical part- or whole-task. For instance, when a trainee is currently performing a tactical maneuver (a context), an instructor would pay attention to specific behavioral cues or performance indicators that are relevant to that maneuver. In many training systems, contexts are usually only implicitly available in the head of an instructors. The CxBR approach can make these contexts explicit and capture performance measurements specific to individual contexts, consequently making these available to an instructional system.

### **Context Determination**

As a computation model, a CxBR model has to determine which context or sub-context is currently active during execution, in order to execute the appropriate decision-making or performance analytics model specific to that context. Active contexts are determined by the context transition rules that can exist between contexts, or by universal transition rules that can be triggered from any context. It is up to the behavior modeler on how to define and implement transition rules, whether they are based on knowledge-based conditions or more advanced classifier algorithms. In conclusion, a well-designed context topology serves as a reference task model for an actor's role behavior, benefiting both the implementation of behavior production and measurement models.

## **TOWARDS INSTRUCTIONAL DESIGN**

In order for CxBR-based HBMs to be used effectively for an adaptive training system, one needs to take into account instructional design aspects. To illustrate this, consider an example of an adaptive simulation-based training system to train fighter pilots in the air combat domain. In the system, human trainees can be trained in a simulator on tactical behaviors such as combat maneuvers, tactical intercepts or specific missions. In the training environment, agent role-players can act as virtual team-members or adversaries. From an instructional design point of view, designing CxBR-based HBMs requires translation from (often) informal, qualitative knowledge used by instructors, to more formal quantitative knowledge to be embedded in an HBM. Below, three of such required translations are discussed.

### **Translating task models to behavior contexts**

Training scenarios are designed around a certain training objective for a trainee (or team of trainees). A training objective can relate to training certain competencies that can be trained through job-specific training tasks in a scenario. Competencies are the knowledge, skills and abilities (KSA) that are required to perform a job well. These

may originate from an established competency model or a competency profile obtained from a training needs analysis (TNA). For instance, competencies for a fighter pilot relate to flying skills (e.g. aircraft handling, anticipation, scan pattern), information handling (e.g. maintaining situation awareness, sensor handling), weapon system handling (e.g. game plan execution, target identification, maneuvering), or communication. Training tasks are so-called part- or whole-tasks that represent concrete (mission-related) activities. Part-tasks are the tasks that can be trained in relative isolation of a mission context, such as basic flight maneuvers (BFMs), take-off/landing or refueling. Whole-tasks group a set of part-tasks that can be trained together, such as a tactical intercept or full missions such as a Defensive or Offensive Counter Air (DCA/OCA). In the design of a CxBR-based HBM, training tasks can be defined as individual contexts and composed in a context topology to represent the behavior in a mission. An example of such a design was given in Figure 2. The tasks can then be individually tracked and measured computationally during a training session.

### **Translating complexity factors to adaptive variables**

From a training perspective, *complexity factors* are the factors that can shape or adapt a training environment in order to adjust the level of complexity of a training session or individual training tasks. In the example domain, complexity factors can relate to environment conditions (e.g. night time, visibility); equipment state (e.g. fuel, weapon load), threats (amount, tactical performance), contingencies (e.g. failures, damages, mission changes) or team-work (e.g. team-members' individual, collaborative and communicative abilities). Some of these factors relate to physical aspects in the environment, whereas other factors relate to cognitive aspects of intelligent role-players. Adaptation requirements for the former need to be translated to adaptive variables that are to be supported by the environment simulation system that is used. Adaptation requirements for the latter should be translated to adaptive variables for CxBR-based HBM models. Behavior modelers can then take them into account in the development in order to prepare the behavior models for real-time adaptation.

### **Translating performance indicators to measurable metrics**

Performance indicators are the qualitative or quantitative measures that are used to judge the performance of a particular training task or competency. In order to measure performance indicators computationally within an HBM, they need to be translated to metrics that can be observed and measured in the simulation. In the example domain of air combat, for some training tasks this can be achieved with relative ease, especially for tasks that are subject to clearly defined rules or protocol, such as defined in the Tactics, Techniques, Procedures (TTPs). For other training tasks, the identification of quantitative metrics can be more challenging for SMEs or instructors, as 'good performance' in complex situations can depend on many contextual factors. For instance, to what extent was a tactical intercept task performed effectively or optimally? In CxBR-based HBM design, performance indicators represented by measurable metrics can be defined for individual contexts, ranging from metrics at the mission level, to whole-task level, to individual part-tasks.

### **Concluding**

Above we have sketched the steps for designing HBMs for adaptive training from an instructional design point of view. In the next section, a proof-of-concept technical demonstrator is described that shows an application of an adaptive training system using CxBR-based HBMs in the air combat domain.

### **PROOF-OF-CONCEPT SYSTEM**

A technical proof-of-concept demonstrator was implemented for an adaptive training system for fighter pilot training. It employs the presented HBM approach for modeling both a human trainee and agent role-players.

In this phase of the study the focus was on the technical infrastructure for the adaptive training system, based on the model described in the beginning of this paper in Figure 1. The demonstrator is built from an existing simulation infrastructure for the air combat domain. The technical framework that was realized is illustrated in Figure 3.

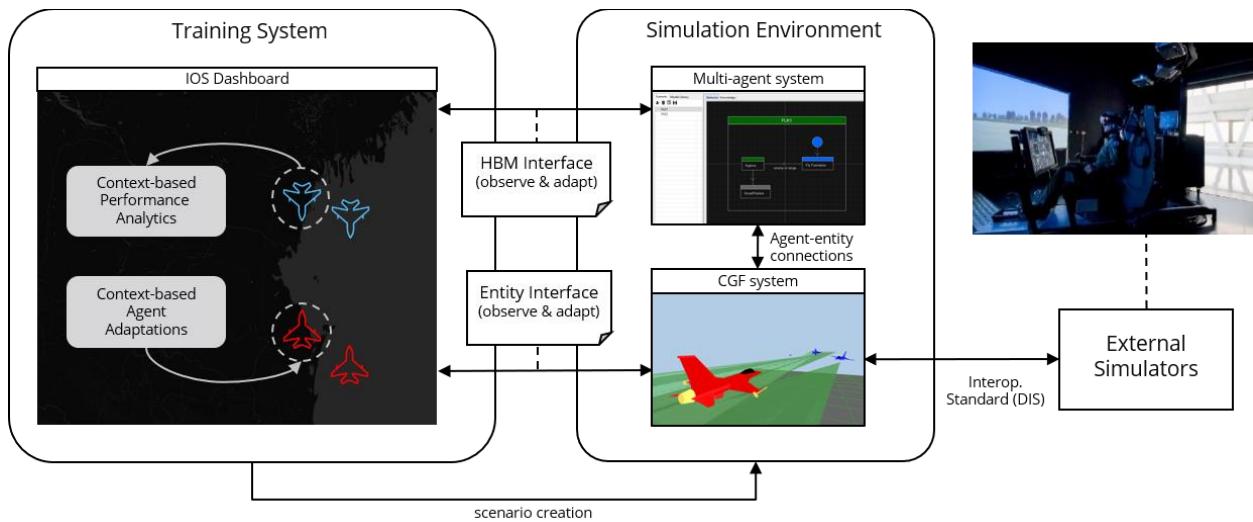


Figure 3: Proof-of-concept Adaptive Training System

The existing simulation infrastructure consists of four (in-house built) components. On the right shows a human trainee interface, representing a research simulator for fighter pilots, designed for tactical training. The simulator connects to the simulation environment (in the middle) through standard interoperability standards such as DIS. The simulation environment consists of two applications: a CGF system (bottom) and a multi-agent system (top). The CGF system is a light-weight air combat simulator, simulating fighter aircrafts with basic platform dynamics, sensor and weapon systems. The multi-agent system is a behavior modeling tool for CGF behaviors. It allows behavior models to control aircrafts in the CGF system (i.e. agent role-players). On the left is the instructional system as an Instructor Operating Station (IOS) that includes functionality for air combat scenario creation, execution and observation through a map-based interface. The system is currently used for fighter pilot and fighter controller training.

For this study we added the following capabilities:

- The multi-agent system was augmented with the capability to develop agents based on the CxBR-based HBM approach. HBMs were developed for the human pilot trainee and agent pilot role-players.
- The IOS was extended to receive real-time contextualized performance measurements from individual HBMs. During training, an instructor can select an aircraft and track current mental states (as contexts) and associated performance metrics, as tracked by the trainee's HBM.
- The IOS was extended to send adaptations to individual agent HBMs. During training, an instructor can select an aircraft and initiate one or more available adaptation for that aircraft.

The above capabilities were demonstrated in a basic air combat scenario: a 2v2 Defensive Counter Air (DCA) mission. Blue force is represented by a human trainee as flight lead, assisted by an agent wingman. Red force is represented by two agent enemy aircrafts. In this scenario, all actors share a comparable HBM design, in line with the CxBR example from Figure 2. The HBMs have contexts for technical-oriented tasks (e.g. formation flying, offensive/defensive maneuvers) and tactical-oriented tasks (e.g. target identification, tactical intercept), based on existing expert models. For each context, several example performance metrics and adaptation variables have been implemented. In the next phase of the study we will focus more on the HBMs' implementation based on expert-validated domain knowledge, concerning task models, complexity factors and performance indicators, following the instructional design steps that were described in the previous section.

The implemented system demonstrates a framework where an adaptive training system has a cognitive-level interface with the intelligent actors in the environment. This allowed operators to inspect and adapt the cognitive environment, as represented by the executing HBMs. This is provided alongside the already existing physical-level interface that allows operators to inspect and adapt the physical environment (e.g. aircraft positions, radar locks or missile positions). While in the current system, a human operator can use these interfaces to evaluate a trainee's performance and plan adaptation, the same interfaces can be used by an automated adaptive instructional system (AIS).

## CONCLUSION

In this paper we presented an HBM approach to support the development of simulation-based, adaptive training in mixed human-agent environments. HBMs in this context are used to model both task behavior models for adaptive agent role-players, and task performance models for trainee actors. As a computational model, we examined the use of the context-based reasoning (CxBR) modeling paradigm to serve as a blueprint for an actor's task model, hereby enabling the implementation of context-specific decision-making and performance models. An HBM can then be employed in a training environment to either represent an agent actor (to drive its behavior), or represent a human trainee actor (to measure its performance). When appropriate HBMs are in place, flexible training environments can be established with interchangeable agent role playing and human trainee actors.

In a simulation-based training environment, the HBMs supports an instructional system by providing a cognitive-level interface with the intelligent actors in the environment: participating HBMs provide real-time observation of trainee performance measurements and expose available adaptations for agents that can be controlled during training for instructional purposes.

In the first phase of this study a technical proof-of-concept implementation of the HBM approach was demonstrated within an existing training system infrastructure for pilot training in the air combat domain. Existing task models were transformed into CxBR-based HBM models that were run in a simulated training environment, and externally interfaced with an instructor dashboard. In following steps we will conduct an instructional-driven design of HBMs by using expert-validated adaptive training scenarios for air combat situations. This allows an evaluation of the effectiveness of the HBMs in an operational training system: either for an instructor-based training system on how it improves trainee assessment and control over adaptive interventions, as for a computer-based AIS on how it supports implementing an instructional strategy.

## REFERENCES

- Abdellaoui, N., Taylor, A., & Parkinson, G. (2009). *Comparative Analysis of Computer Generated Forces' Artificial Intelligence*.
- Arar, Ö. F., & Ayan, K. (2013). A flexible rule-based framework for pilot performance analysis in air combat simulation systems. *Turkish Journal of Electrical Engineering and Computer Sciences*, 21(8), 2397–2415.
- Bell, B., Nye, B., Bennett, W., & Kelsey, E. (2021). Attention and Engagement in Virtual Environments: Measuring the Unobservable. *Interservice/Industry Training, Simulation and Education Conference (IITSEC)*.
- Bell, B., & Sottilare, R. (2019). Adaptation Vectors for Instructional Agents. In R. A. Sottilare & J. Schwarz (Eds.), *Adaptive Instructional Systems* (pp. 3–14). Springer International Publishing.
- Castro, J., Gonzalez, A. J., & Gerber, W. J. (2002). Design and Implementation of CITKA, a Context Based Tactical Knowledge Acquisition System. *Swedeng American Workshop on Modeling and Simulation-2002*.
- Dong, Y., Ai, J., & Liu, J. (2019). Guidance and control for own aircraft in the autonomous air combat: A historical review and future prospects. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(16), 5943–5991.
- Doyle, M. J., & Portrey, A. M. (2014). Rapid adaptive realistic behavior modeling is viable for use in training. *Proceedings of the 23rd Conference on Behavior Representation in Modeling and Simulation (BRIMS)*, 73–80.
- Freeman, J., Watz, E., & Bennett, W. (2020). Assessing and Selecting AI Pilots for Tactical and Training Skill. *NATO MSG-177*.
- Freeman, J., Watz, E., & Bennett, W. (2019). Adaptive Agents for Adaptive Tactical Training: The State of the Art and Emerging Requirements. In R. A. Sottilare & J. Schwarz (Eds.), *Adaptive Instructional Systems* (pp. 493–504). Springer International Publishing.
- Lewis, C. L., M. Alexander, T. Huiskamp, W. , Blais. (2019). *MSG-127 A Reference Architecture for Human Behaviour Representation*. NATO.
- Lotens, W., Allender, L., Armstrong, J., Belyavin, A., Cain, B., Castor, M., Gluck, K., Käppler, W., Kwantes, P., Lundin, M., Thomas, G., & Wallin, N. (2009). *HFM-128 Human Behavior Representation in Constructive Simulation*. NATO.
- Mansikka, H., Virtanen, K., Harris, D., & Jalava, M. (2021). Measurement of team performance in air combat – have we been underperforming? *Theoretical Issues in Ergonomics Science*, 22(3), 338–359. <https://doi.org/10.1080/1463922X.2020.1779382>

- Mohanavelu, K., Poonguzhali, S., Adalarasu, K., Ravi, D., Chinnadurai, V., Vinutha, S., Ramachandran, K., & Jayaraman, S. (2020). Dynamic cognitive workload assessment for fighter pilots in simulated fighter aircraft environment using EEG. *Biomedical Signal Processing and Control*, 61, 102018.
- Portrey, A. M., Keck, L. B., & Schreiber, B. T. (2006). *Challenges in developing a performance measurement system for the global virtual environment*. Lockheed Martin Systems Management MESA AZ.
- Rowe, L. J., Prost, J., Schreiber, B., & Bennett Jr, W. (2008). Assessing High-Fidelity Training Capabilities Using Subjective and Objective Tools. *2008 Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) Proceedings*.
- Salas, E., Rosen, M. A., Held, J. D., & Weissmuller, J. J. (2009). Performance Measurement in Simulation-Based Training: A Review and Best Practices. *Simulation & Gaming*, 40(3), 328–376. <https://doi.org/10.1177/1046878108326734>
- Sottilare, R., & Brawner, K. (2018). Exploring standardization opportunities by examining interaction between common adaptive instructional system components. *Proceedings of the First Adaptive Instructional Systems (AIS) Standards Workshop, Orlando, Florida*.
- Stensrud, B. S., Barrett, G. C., & Gonzalez, A. J. (2004). Context-Based Reasoning: A Revised Specification. *FLAIRS Conference*, 603–610.
- Stensrud, B. S., Barrett, G. C., Lisetti, C. L., & Gonzalez, A. J. (2002). Modeling Affect in Context-Based Reasoning. *Proceedings of the Swedish-American Workshop on Modeling and Simulation (SAWMAS)*.
- van den Bosch, K., Blankendaal, R., Boonekamp, R., & Schoonderwoerd, T. (2020). Adaptive Agents for Fit-for-Purpose Training. In C. Stephanidis, D. Harris, W.-C. Li, D. D. Schmorow, C. M. Fidopiastis, P. Zaphiris, A. Ioannou, X. Fang, R. A. Sottilare, & J. Schwarz (Eds.), *HCI International 2020 – Late Breaking Papers: Cognition, Learning and Games* (pp. 586–604). Springer International Publishing.
- van Oijen, J., Toubman, A., & Claessen, O. (2021). Teaching Reinforcement Learning Agents with Adaptive Instructional Systems. In R. A. Sottilare & J. Schwarz (Eds.), *Adaptive Instructional Systems. Design and Evaluation* (pp. 120–136). Springer International Publishing.

## Research Article

# Research on English Vocabulary and Speech Corpus Recognition Based on Deep Learning

Wang Zhen 

*Department of Public Education, Inner Mongolia Technical College Of Construction, Hohhot 010070, China*

Correspondence should be addressed to Wang Zhen; b20160904105@stu.ccsu.edu.cn

Received 8 June 2022; Revised 14 August 2022; Accepted 22 August 2022; Published 19 September 2022

Academic Editor: Jun Ye

Copyright © 2022 Wang Zhen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to investigate how to recognize English words and speech corpus, an English vocabulary and English speech recognition model based on deep learning algorithm was proposed. Through recommending key technical problems and solutions based on deep learning algorithm, how to realize the recognition of English vocabulary and speech corpus was investigated. In the research, the accuracy of the method on the English vocabulary and speech corpus recognition based on the deep learning algorithm increased 79% over the previous methods. Combined with the principle of the deep automatic encoder and deep learning algorithm, the research emphasis was on the effects of speech recognition framework for speech corpus. The speech recognition research based on the theory of deep learning not only had a theoretical guidance meaning but also had the use value in the practical application.

## 1. Introduction

Due to the complex changes in speech pronunciation, the large amount of data of speech signals, the high dimension of speech feature parameters, and the large amount of computation for speech recognition and evaluation, high-demand software and hardware resources and algorithms are required for large-scale speech signal processing. However, the traditional speech recognition algorithm dynamic time warping algorithm, hidden Markov model, and artificial neural network have their own advantages and disadvantages, they have encountered unprecedented bottlenecks, and it is difficult to further improve their accuracy and speed. In recent years, with the development of deep learning research in the field of machine learning and the accumulation of big data corpus, speech recognition and evaluation technology has developed rapidly. Language is an essential element in people's daily communication. Speech is also an essential means of information exchange in people's daily life and work. Clear speech expression can further clarify the expression of information and further simplify the main idea that people want to express. In addition, a large piece of relatively complex information can be

decomposed into different parts to help people better communicate and analyze [1]. But with the development of information technology and Internet technology, the emergence and development of speech recognition has gradually changed the people's living habits. By using intelligent terminal, computer, intelligent wear equipment, speech recognition can be realized. Speech recognition technology has become one of the technical means of language communication in today's society. How to better apply this technology to assist people's communication is the focus of research [2]. Therefore, combining the principle of deep autoencoder and deep learning algorithm, an English vocabulary and English speech recognition model based on deep learning algorithm is proposed, which focuses on the influence of speech recognition framework on speech corpus.

Speech recognition technology was mainly divided into the following stages. In 1950s, Audrey in AT&T Bell Laboratory was the prototype of speech recognition. In late 1960s and early 1970s, it has a significant progress [3]. In late 1980s, it has a breakthrough. In early 1990s, many large companies launched their own speech recognition apps. The Audrey system, first developed at AT&T Bell Labs in the 1950s, was the first speech recognition system capable of

recognizing 10 English digits. However, substantial progress was made in the late 1960s and early 1970s [4]. The main reason was the introduction of linear predictive coding plane (LPC) technology and dynamic time warping (DTW) technology, which could effectively solve the problem of feature extraction and unequal length matching of speech signal. Speaking skills at that time were generally based on the principle of template matching. And the name is limited to identifying the difference between special people and small words. The identification of specific population segregation based on cepstral prediction and DTW techniques has been observed. At the same time, vector quantization (VQ) and hidden Markov model (HMM) theory were proposed [5].

In the late 1980s, lab speech recognition research finally had a breakthrough. For the first time in the lab, the barriers of large vocabulary, continuous speech, and nonspecific people were by combining all three characteristics in one system, typically Carnegie Mellon University's Sphinx system. It was the first high-performance nonspecific large vocabulary continuous speech recognition system. During this period, speech cognition research was further understood, characterized by the use of HMM models and neural network devices in speech cognition [6]. Zhang et al. at AT&T Bell Labs evaluated the HMM model for a wide range of applications. They engineered the original difficult HMM pure mathematical model, so that more researchers could understand it and make statistical methods that will become the mainstream of speech recognition technology. Statistical methods shifted researchers' attention from micro to macro. They no longer deliberately pursued refinement of speech features but constructed the best speech recognition system more from the overall average (statistical) perspective [7]. The research on speech recognition in China started in the 1950s. The Institute of Acoustics of Chinese Academy of Sciences began to conduct speech research. The real beginning of speech recognition in China should be the generation of RTSRS(01), a speech recognition system which used bandpass filter bank parameters and realized by Institute of Acoustics of Chinese Academy of Sciences in 1978. In the 1980s, professors from Tsinghua University proposed an implicit Markov model based on segment state distribution, which effectively solved the pruning problem of language identification in multilingual continuous recognition system [8]. In 2004, some scholars used HMM and GMM to score Chinese pronunciation and tone, respectively. Downhill Simplex Search optimized subsystem parameters in order to achieve the same scoring standard consistent with Chinese experts. Fluent application systems included FLUENCY of Language Technology Research Institute of Carnegie Mellon University and School of Information of Kyoto University in Japan. Some institutions in China, such as PLASER at Hong Kong University of Science and Technology, Department of Electronic Engineering at Tsinghua University, Department of Computer Science of Harbin Institute of Technology, and the Department of Computer Science of Harbin Institute of Technology, have also made some significant progress in these researches. However, most researches in

China were to assist the learning of Chinese pronunciation [9].

## 2. Methods

### 2.1. Key Technologies of Deep Learning

**2.1.1. Energy Probability Model.** Introducing RBM into network modeling is a breakthrough with theoretical guiding significance for deep neural networks [10]. Using RBM as an energy model, it is possible to model arbitrarily distributed data. The Boltzmann machine is a large class of neural network models, but the most commonly used one in practice is the RBM. The RBM itself is simple, just a two-layer neural network, so it is not strictly considered as a category of deep learning. When the minimum energy of the overall network is calculated iteratively, it means that the system is in steady state at this time and the network parameters we require are also the network parameters at this time.

**2.1.2. Pretraining Layer by Layer.** In the past, neural networks determined the initial value through random initialization, at which time the random option value was required. It was often inconsistent with the actual situation, so the final effect was not ideal [11]. With RBM, the model is built in the middle of the two adjacent layers, and the training is carried out layer by layer from bottom to top. After several iterations, RBM enters a relatively stable state. Each neuron in the visible layer is connected to all the neurons in the hidden layer, but there is no connection between the neurons in the same layer, and all the neurons have only two output states. In this case, the hidden layer and the visible layer are equivalent to the same features in more than one space in different expressions, so as to determine the initial value consistent with the weight of the actual situation [12].

**2.1.3. Network Parallel Training.** Considering that there are several hidden layers in the deep neural network, each of which has more than 1000 nodes, the number of relevant parameters is likely to exceed 1 million. In such a large-scale network, the time of data training will be greatly extended without parallel processing. The BP neural network is mainly composed of the input layer, the hidden layer, and the output layer. The number of nodes in the input and output layer is fixed. Whether it is a regression or a classification task, choosing the appropriate number of layers and the number of hidden layer nodes will affect the performance of the neural network to a large extent. Parallel processing can be completed by hardware or software. The hardware method requires the support of GPU or distributed computing cluster. The software method means that the parameters of data subset are updated by multithreading and the updated results are unified at an appropriate time to complete the parallel training of the network [13].

### 2.2. Encoder Category Based on Depth Theory

**2.2.1. Deep Autoencoder.** The input required by the deep autoencoder is the original data feature. And the middle

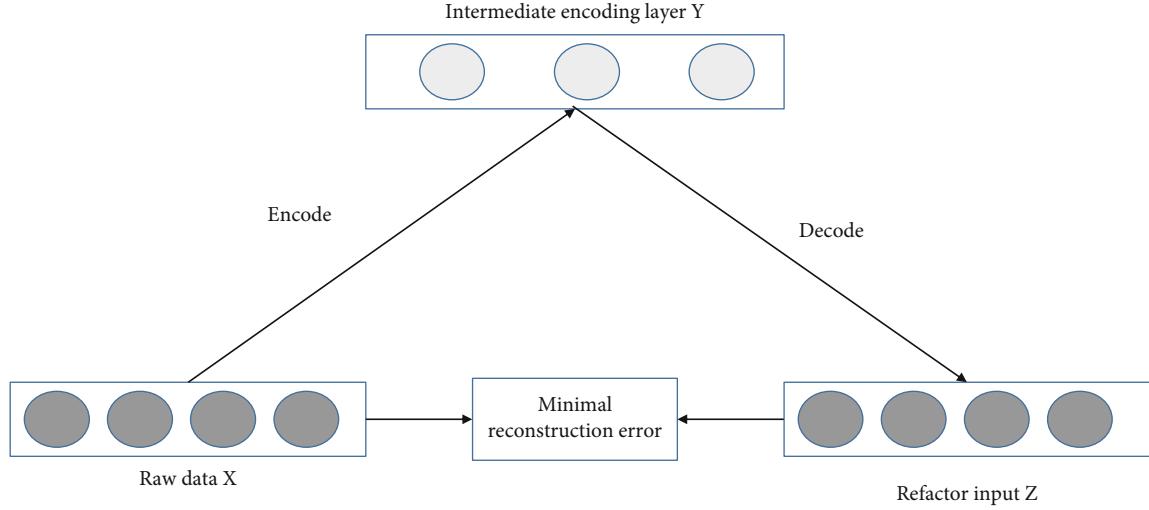


FIGURE 1: Autoencoder model.

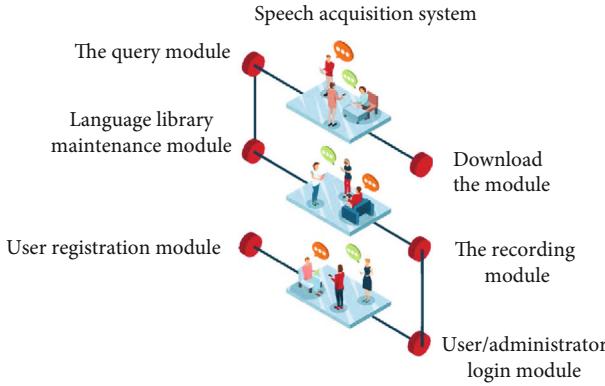


FIGURE 2: Speech acquisition system module.

layer encoding feature is obtained by different hidden layer encoding and the original input is reconstructed according to the decoding. Autoencoder is a kind of neural network, whose basic idea is to directly use one layer or more layer of neural network to map the input data and get the output vector. The model is shown in Figure 1. Network parameter adjustment is mainly aimed at minimizing the mean square error between original input and reconstructed input. The calculation method of loss function is shown in

$$J(W, b) = \left[ \frac{1}{m} \sum_{i=1}^m J(W, b, x^{(i)}, y^{(i)}) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_i-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2, \quad (1)$$

$$J(W, b) = \left[ \frac{1}{m} \sum_{i=1}^m \left( \frac{1}{2} \|x^{(i)} - h_{W,b} x^{(i)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_i-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2. \quad (2)$$

In formula (1) and formula (2), the first term represents average reconstruction error. The second term represents regularization constraint term, aiming to prevent overfitting [14].  $m$  represents the amount of training data.  $W$  and  $b$  are

parameters of the encoder.  $x^{(i)}$  and  $y^{(i)}$  represent the original input and reconstruction input in turn, and their relationship is shown in

$$y^{(i)} = h_{W,b}(x^{(i)}). \quad (3)$$

**2.2.2. Denoising Autoencoder.** The training data required for this encoder is random noise that is superimposed on the raw data before providing it to the network (adding random noise to input layer nodes or according to some probability to make some input layer nodes 0). After the coding module is used to obtain the coding representation of the middle layer, the original data is reconstructed on the output layer to obtain more prominent features in robustness. The original data layer (the data in it is the raw data, without any processing) is the original json format data, because the original data has two kinds of data: start log and event log.

**2.2.3. Sparse Autoencoder.** Sparse autoencoder, another important extension model of autoencoder, also has good feature extraction performance. Sparse means that the hidden layer node has a high probability of 0 and sparse autoencoder is an unsupervised machine learning algorithm that constantly adjusts the parameters of the autoencoder by calculating the error between the autoencoding output and the original input to finally train the model. Autoencoders can be used to compress the input information and extract useful input features, and its non-0 time is relatively short (there is a long distance between it and 0; that is, it is in active state) [15]. Research on the visual perception system of human brain shows that the distribution of visual cortex cells in V1 region is sparse after the human brain receives natural image signals, even though only a few of them are activated at the same time. The output state of the hidden layer of the network is limited, so that the nodes of the hidden layer enter the sparse state, and the average output of the nodes of the hidden layer is equal to 0. In this way, the proportion of active nodes is relatively small, and the homogeneity of

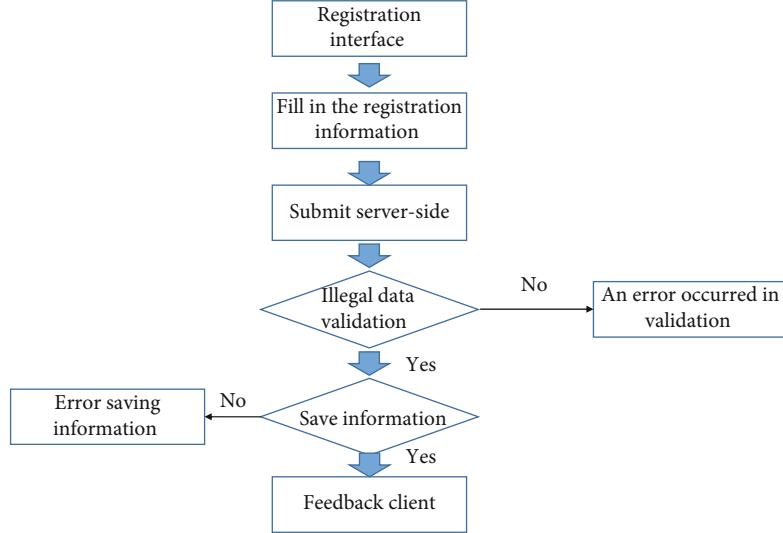


FIGURE 3: User registration process.

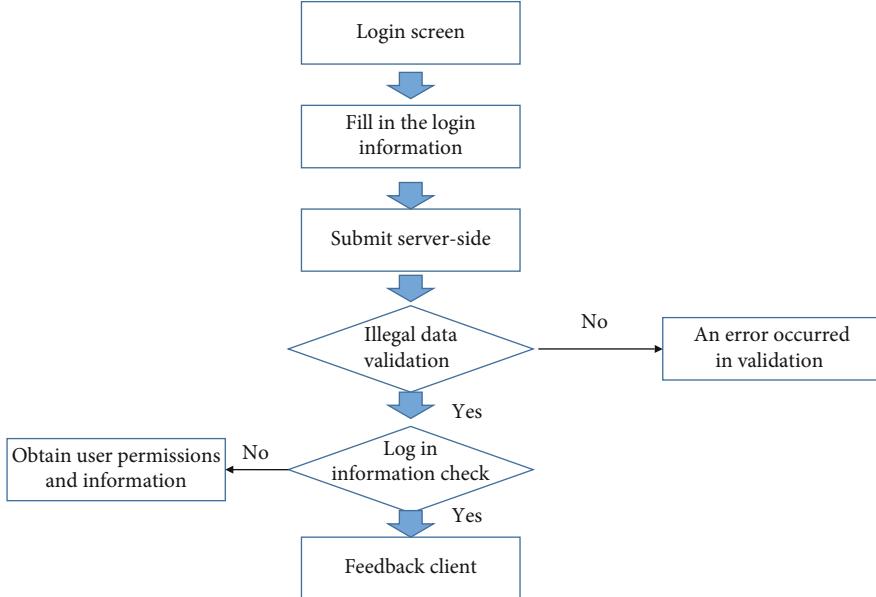


FIGURE 4: Login module process.

the characteristics of the nodes of the hidden layer will not occur [16]. The loss function of sparse autoencoder is shown in

$$J_{\text{sparse}}(W, b) = J(W, b) + \beta \sum_{j=1}^{s_2} \text{KL}\left(\rho \parallel \bar{\rho}_j\right). \quad (4)$$

The first term of formula (4), which is the same as formula (1), represents the size of reconstruction error. The second term is KL distance, representing the gap between the expected sparsity and the actual value, which can be cal-

culated by the following expression, as shown in

$$\text{KL}\left(\rho \parallel \bar{\rho}_j\right) = \rho \log \frac{\rho}{\bar{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \bar{\rho}_j}. \quad (5)$$

$\bar{\rho}_j$  represents the average output value of nodes at the hidden layer, which satisfies

$$\bar{\rho}_j = \frac{1}{m} \sum_{i=1}^m \left[ a_j^{(2)} \left( x^{(i)} \right) \right]. \quad (6)$$

### 2.3. System Framework Based on Deep Autoencoder

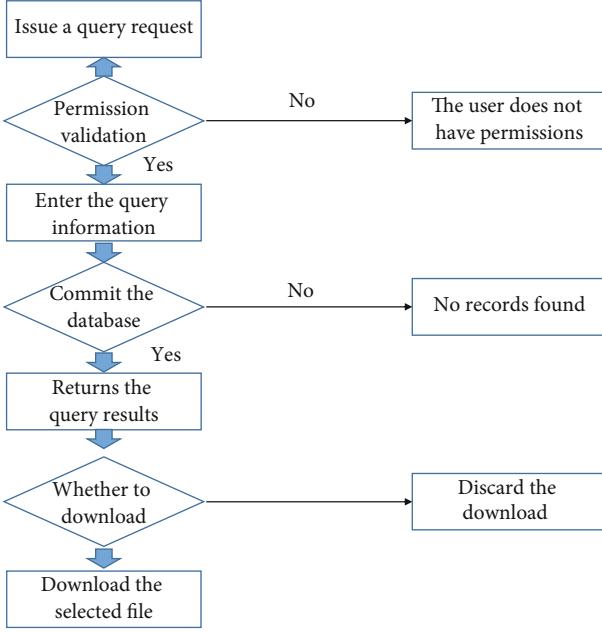


FIGURE 5: Query and download module flow.

**2.3.1. Experimental Corpus.** The original data required in the experiment are all from TIMIT speech data set. The full name of TIMIT is The DARPA TIMIT Acoustic-Speech Continuous Speech Corpus, which is collected and constructed by Texas Instruments, Massachusetts Institute of Technology, and Stanford Research Institute. There are 6,300 sentences sampled at 16 kHz from eight different locations in the United States, and all sentences are manually segmented and labeled.

**2.3.2. Feature Preprocessing.** In the process of extracting high-level features, deep neural networks generally need to receive acoustic features such as MFCC and Fbank. Because of the copronunciation phenomenon, it is necessary to extract digital features from images (or texts) for use by various models. Sometimes, you need to extract numerical features from images (or text) for use by various models. Deep learning models can be used not only for classification regression but also for extract features. The trained model is usually used to input pictures and output as extracted feature vectors. It is generally necessary to expand the short-term features to obtain the superframe features carrying context information. Original feature extraction is as follows: according to the parameters of frame length 20 ms and frame shift 10 ms, the 39-dimensional MFCC features (12-dimensional output + 1-dimensional logarithmic energy and their first- and second-order differences) are extracted from the original speech through the HCopy file provided by HTK. A voice sample in the data for detailed description is selected. First, two text files should be created in the same root directory, named YL.conf and YL.scp, respectively. The former mainly writes parameters for MFCC extraction, and the latter is the path of sample files and generated files. The yangli.mfc file can be obtained in the same directory after the extraction is successful. Since the file format cannot

be directly viewed, the HList tool can be used to convert it to a txt file.

Data preprocessing is as follows: 5 frames are added before and after the features obtained in the previous step to obtain 11 consecutive superframe features. Then, the cepstrum mean variance is normalized. The processed features are input through the visibility layer as training samples of the network model [17]. In the process of normalization of each dimension of superframe feature, the two points cannot be ignored. First, normalization can reduce the influence caused by feature difference between channel and individual. Second, Gauss-Bernoulli RBM model is selected in the process of modeling the input layer and the first hidden layer whose node states conform to Gaussian distribution. At this time, the energy function is shown in

$$E(v, h) = \sum_{i \in V} \frac{(v_i - a_i)^2}{2\sigma_i^2} + b^T h - \sum_{i \in V, j \in H} \frac{v_i}{\sigma_i} h_j w_{ij}. \quad (7)$$

After CMVN processing, input data distribution in formula (7) satisfies

$$\begin{cases} a_i = 0, \\ \sigma_i = 1. \end{cases} \quad (8)$$

The energy function is equivalent to

$$E(v, h) = \sum_{i \in V} \frac{v_i^2}{2} - b^T h - \sum_{i \in V, j \in H} v_i h_j w_{ij}. \quad (9)$$

**2.3.3. Autoencoder Structure.** The structure of the encoder includes the number of hidden layers, the number of nodes contained in each hidden layer, and the node type of each hidden layer [18]. After many experiments, the deep autoencoder used in the study consists of seven layers, including an input layer, an output layer, and five hidden layers, and the number of nodes in each layer is  $490 \times 720 \times 720 \times 50 \times 720 \times 720 \times 490$  [19].

#### 2.3.4. Network Training Algorithm

(1) **Gauss-Bernoulli RBM Training.** Because the network input has the speech cepstrum feature, the value is between  $[-\infty, +\infty]$ , which is obviously different from the black and white image signal. Gauss-Bernoulli RBM is often selected as the input layer and the first hidden layer to build the model. In practice, data preprocessing is needed to normalize the input feature mean and variance. The first several layers of the model are mainly divided into visible layer (490 Gauss nodes) and hidden layer (720 Bernoulli nodes). Here is the training algorithm.

- (1) Given a sample  $v$  of training data, the activation probability of hidden layer node  $h_j$  can be expressed as shown in

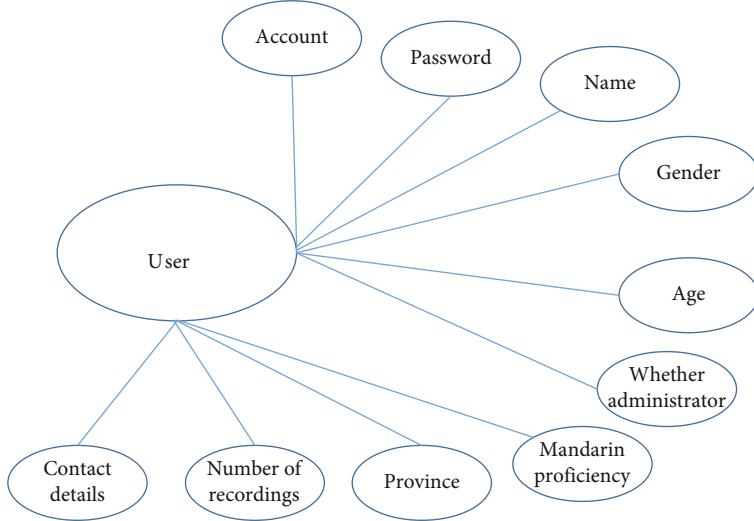


FIGURE 6: User entity attribute diagram.

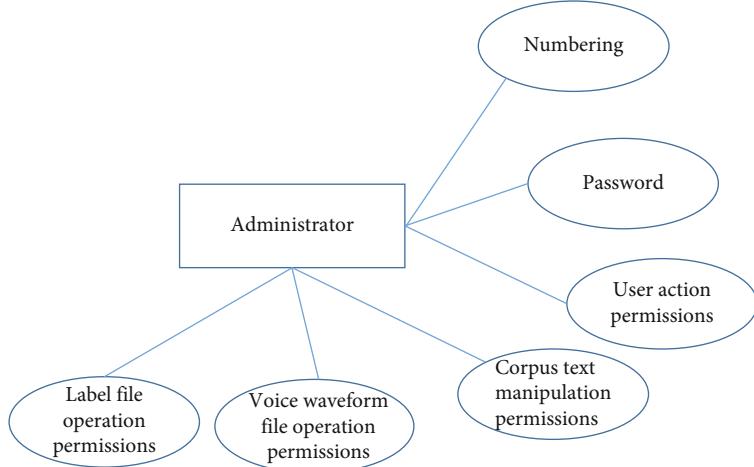


FIGURE 7: Administrator entity diagram.

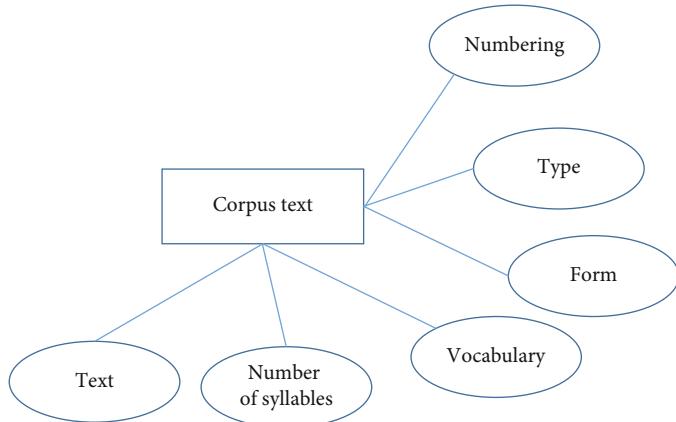


FIGURE 8: Corpus text entity diagram.

$$p(h_j = 1 | v) = \sigma \left( b_j + \sum_{i \in \text{vis}} v_i w_{ij} \right) \quad (10)$$

- (2) Randomize the hidden layer node values obtained in (1) to generate 0 and 1 activation states, and deduce the visible layer input  $v'$  according to the hidden layer node states. For the linear visible layer element, its reconstruction formula is expressed as

$$v' = N \left( b_i + \sum_{j \in \text{hid}} h_j w_{ij}, 1 \right) \quad (11)$$

- (3) The reconstructed visible layer state value  $v$  is used as the input of RBM structure. The hidden layer probability  $h$  is calculated again according to step (1)  
 (4) Update weight parameters according to formula (12), where  $\langle \cdot \rangle$  is the average value of all samples in each small batch and  $\epsilon$  is the learning rate, as shown in

$$\Delta w_{ij} = \epsilon \left( \langle v_i h_j \rangle - \langle v'_i h'_j \rangle \right) \quad (12)$$

The initialization parameters of the model are as follows: the weight of the connection is set to a small value and the node bias is set to 0. When each size is done, there are 256 minibatch models. Degrees are 0.01. The activation probability value of each node of the last training hidden layer  $h1$  is retained as the input data of visible layer of RBM in the upper-middle layer of superposition structure [20].

(2) *Bernoulli-Bernoulli RBM Training.* The output value of the hidden layer of the first Bernoulli-Bernoulli RBM model is directly defined as the input value of the visible layer of the next RBM, and then, the connection weight between  $h1$  and  $h2$  of the hidden layer is continued to be trained. Compared with Gauss-Bernoulli RBM, the training method is basically similar, but the visible layer nodes obey Bernoulli distribution. Here, the hidden layer state is used to reconstruct the visible layer, and the basic formula is shown in

$$p(v'_i = 1 | h) = \sigma \left( b_i + \sum_{i \in \text{hid}} h_i w_{ij} \right). \quad (13)$$

(3) *Network Parameter Tuning.* After the initial model is pretrained, network parameters need to be adjusted, usually through backpropagation (BP) [21]. The sample overall loss

function can be written as shown in

$$J(W, b) = \left[ \frac{1}{m} \sum_{i=1}^m J(W, b, x^{(i)}, y^{(i)}) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} \left( W_{ji}^{(l)} \right)^2, \quad (14)$$

$$J(W, b) = \left[ \frac{1}{m} \sum_{i=1}^m \left( \frac{1}{2} \|h_{W,b}(x^{(i)}) - y^{(i)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} \left( W_{ji}^{(l)} \right)^2. \quad (15)$$

The first term of formula (14) is the mean square deviation term, which reflects the degree of difference between reconstruction and original input features. The second term is added to avoid the overfitting problem, which is the so-called regularization term. The contribution of the first and second terms to the loss function can be balanced by increasing the weight attenuation parameter  $\lambda$ .  $h_{W,b}(x^{(i)})$  represents the reconstruction result obtained through the process of coding and decoding the sample  $x^{(i)}$  through the network. The present invention provides a research on the recognition of English vocabulary and speech corpus based on a deep learning algorithm, which comprehensively evaluates the English pronunciation quality of the preset object through the two different aspects of the English pronunciation and the English vocabulary, so that it can comprehensively evaluate the English pronunciation quality of the preset object. It can accurately evaluate the actual English pronunciation accuracy and standard degree of the preset object and give an objective and reliable pronunciation quality evaluation score accordingly, so as to effectively improve the English pronunciation quality and improve the experience of learning English.

### 3. Results and Analysis

The design of the overall structure of the system is to reasonably divide the whole system into various functional modules, so as to correctly handle the relationship between and within modules, as well as the data connection between them, and then to define the internal structure of each module [22].

*3.1. Structural Design of the System.* In the system, C/S system architecture is used, including five function modules, namely, user registration module, user administrator login module, recording module, database maintenance module, and query and download module (see Figure 2).

#### 3.2. Detailed Design

##### 3.2.1. System Module

*(1) User Registration Module.* The registration module realizes the user registration function and corresponds the recording information with the account, which is convenient for users to query and use in the future. Account number, password, age, gender, Mandarin proficiency, and native place will be written into the user information form as

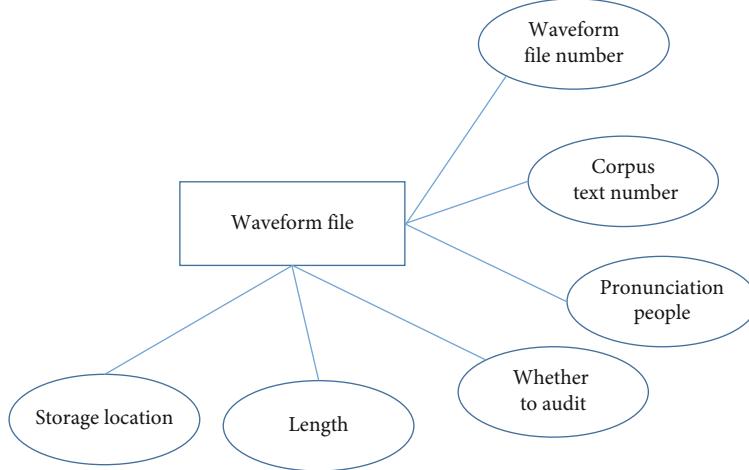


FIGURE 9: Pronunciation waveform file entity diagram.

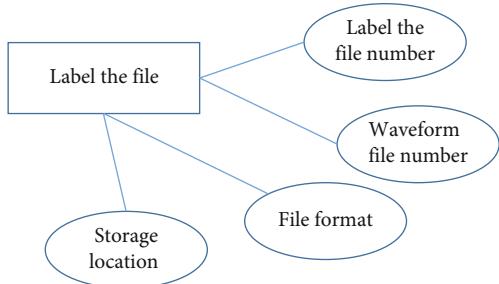


FIGURE 10: Labeling the file entity diagram.

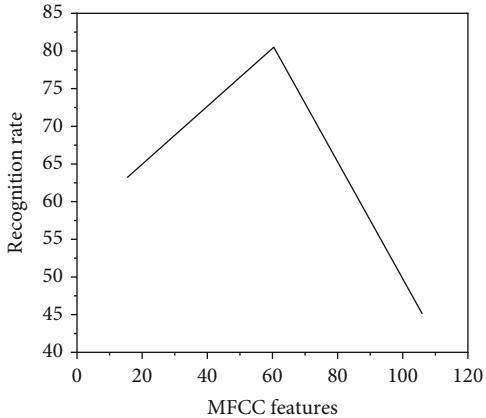


FIGURE 11: Comparison of recognition rates between MFCC features and DAE midlayer features (broken lines).

required. If the registration fails, the system prompts you to review the registration information and returns to the registration page. Its functional flow chart is shown in Figure 3.

(2) *User Administrator Login Module.* The login module ensures that users can log in to the system with legitimate identities and obtain the recording information for easy query and modification. First, the user fills in the account number and password as parameters and passes them to

the server, which is compared with the information in the user information table [23]. If the authentication fails, an error message is displayed and the registration page is displayed. If the authentication succeeds, the user information and permission are displayed. Its function flow chart is shown in Figure 4.

(3) *Recording Module.* Recording module includes a recording program. After the program receives the user's request, according to the user's choice, the corresponding recording text is selected. The second step is to initialize the recording device on the machine and start recording. After the recording is complete, the user information, recording files, and text information are sent back to the server as parameters, and the save program is invoked for further processing.

(4) *Database Maintenance Module.* Database maintenance module is used to operate the database for administrators. The client provides an exchange interface, which is convenient for administrators to log in so as to manage and audit the user, corpus text, waveform files, and annotated files. The first step is to identify verification to see whether they have the authority to manage the database. After passing the verification, they can operate and manage the database and save the process of modifying information.

(5) *Query and Download Module.* The user sends a query request to confirm the user permission. Then, the query information input by users (articulator attributes, corpus text keywords, waveform file numbers, etc.) is transferred to the server as parameters. And the server returns the query result, and the user can select the corresponding file according to the returned result for download. Its function flow chart is shown in Figure 5.

3.2.2. *The Conceptual Design of Database.* The goal of the conceptual design is to accurately describe the information schema of the application domain and support the various applications of the user, so that it is easy to transform into database logic schema and easy to understand by the user. The typical method of conceptual model design is E-R

method. A diagram is composed of three parts, including entity, attribute, and connection. According to the requirement analysis of the system function and database described above, the E-R diagram of the system can be obtained, as shown in Figures 6-10.

### 3.2.3. System Implementation

(1) *Implementation of Landing Module.* In the process of system design, no matter if it is divided into several modules and different modules, its operators are different. The foundation of any successful application's security policy is a robust means of authentication and permission control and secure communications that provide data integrity and confidentiality. The design of login module is mainly to verify the correctness of user account and password.

(2) *Implementation of Registration Module.* It is mainly used for the personal information of registered users. Each user can view and modify their own information, as well as refer to previous personal recording records.

(3) *Implementation of Recording Module.* The first step is to initialize the recording device on the local device, and then, start recording after the user selects the corresponding text. After recording, the recording file is saved to the server.

(4) *Implementation of Language Library Maintenance Module.* It is mainly to achieve the management of the database, including the management of users, corpus text, waveform file management, and labeling file management.

(5) *Implementation of Query and Download Module.* Through the search function, users can find the corresponding text corpus, waveform files, and annotated files and download the required files.

Taking the tagging of speech corpus as an example, a complete speech corpus should not only contain original speech data and corresponding pronunciation text but also corresponding label files. In order to improve the utilization value of speech corpus, the key is to label the speech corpus completely. Corpus refers to a large-scale electronic text library scientifically sampled and processed. With the help of computer analysis tools, researchers can carry out relevant language theory and applied research. The label process of speech corpus is a process of language knowledge formalization. The label quality and depth of the speech corpus directly affect the accuracy and richness of information mined from the speech corpus and determine the availability and value of the speech corpus to a great extent. Based on statistical principles, we can find the habitual collocation of language, and the corpus can help us to better master the language. Is the tool for our research and collocation. A complete label system is a very important part of corpus construction, and the complete label includes segmenting and prosodic label. The so-called English phonetic segment annotation is to segment each phonetic unit (sentence, word, character, syllable, consonant, and vowels) in a continuous speech stream and describe their timbre characteristics, mainly including vowels, consonants, and combinations of vowels and vowels, vowels and consonants, and consonants and consonants.

A GMM-HMM acoustic model is simultaneously trained on HTK platform for the two features (unsupervised and supervised) and MFCC features obtained through deep autoencoder model training. The recognition accuracy of words and sentences is used as the experimental comparison results, as shown in Figure 11.

## 4. Conclusions

As one of the hottest research fields at present and in the future, deep learning has achieved good results in the field of speech recognition. The performance of speech recognition system often directly affects the effect experience of most intelligent systems, so the future development direction must be to combine the two technologies to promote mutual progress. Based on the theory of deep learning, the research comprehensively discusses the application value and effect of deep learning model in the field of speech recognition, starting from speech feature extraction and acoustic modeling.

- (1) Taking acoustic feature extraction as the research object, the research work was carried out based on the deep autoencoder model. Deep autoencoders belonged to multilayer network model and were widely used in data dimension reduction and feature extraction based on unsupervised training. The research focused on the analysis of the deep learning model from the perspectives of feature data preprocessing, model structure, and network training parameters. The automatic encoder was established based on the speech features in MATLAB platform to extract the new speech features from the original MFCC features. Finally, HTK recognition tool was used to test and verify the TIMIT English speech corpus. Compared with the original situation, the new features extracted from the unsupervised and supervised training improved the English word recognition rate by 1.64% and 2.86% and the English sentence recognition rate by 2.55% and 6.53%, respectively
- (2) Taking acoustic modeling as the research object, the research work was carried out based on DNN-HMM. As a discriminative model, deep neural network was applied in the field of acoustic modeling. It relied on the output layer to represent the HMM state output probability. And with the help of its own network structure, it could meet the requirements of complex feature modeling. It replaced the original GMM model and combined with HMM to obtain the acoustic model based on DNN-HMM. The acoustic models based on the GMM-HMM and DNN-HMM were modeled by Kaldi speech recognition system platform. Finally, experiments on TIMIT speech corpus prove that compared with

the GMM-HMM model, the English word recognition error rate and sentence recognition error rate of DNN-HMM model were reduced by 30.3% and 17.2%, respectively.

Currently, with the rapid development of computer technology, English vocabulary and speech corpus recognition technology have also obtained the rapid development. And there are more and more technologies applied to the actual products, such as speech input system and computer assisted language learning system. Products are constantly emerging, which provides superior service for the people. For an excellent speech synthesis and recognition system, a speech corpus with high information content and low redundancy is essential. It can be seen that speech corpus plays an important role in speech recognition, speech synthesis, and other areas of speech research.

In the research, an English vocabulary and speech corpus was proposed and built to expand the sources of speech corpus and improve the efficiency of English vocabulary and speech corpus recognition and synthesis system construction. The following work were mainly completed.

- (1) For English vocabulary and speech corpus selection, the original corpus text was automatically downloaded from the Internet firstly, and then, the greedy algorithm was used to screen the original corpus (based on high frequency words and three-tone words), and the final recorded corpus text was obtained
- (2) For speech recording and corpus management system, the recording module working process and the design idea were described in detail. And the speech database management system was implemented, which was convenient for user to operate text, audio files, and tagging corpus query and download files. The recording work was tested, and the English vocabulary and speech corpus was established
- (3) For speech file label, the speech corpus label standards and guidelines for corpus label in the United States were introduced. And then, preliminary speech label files were automatically generated by the program without alignment, which could effectively reduce the workload. And then through the software, the manual alignment work was performed and the final label files were obtained

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The author declares that there are no conflicts of interest.

## References

- [1] V. H. Vu, Q. P. Nguyen, K. H. Nguyen, J. C. Shin, and C. Y. Ock, "Korean-Vietnamese neural machine translation with named entity recognition and part-of-speech tags," *IEICE Transactions on Information and Systems*, vol. 103, pp. 866–873, 2020.
- [2] D. Lemmenmeier-Batinić, "Converting raw transcripts into an annotated and turn-aligned TEI-XML corpus: the example of the corpus of Serbian forms of address," *Slovenščina 20 Empirical Applied and Interdisciplinary Research*, vol. 9, no. 1, pp. 123–144, 2021.
- [3] K. Zvarevashe and O. O. Olugbara, "Recognition of speech emotion using custom 2d-convolution neural network deep learning algorithm," *Intelligent Data Analysis*, vol. 24, no. 5, pp. 1065–1086, 2020.
- [4] L. R. Kishline, S. W. Colburn, and P. W. Robinson, "A multi-media speech corpus for audio visual research in virtual reality (I)," *The Journal of the Acoustical Society of America*, vol. 148, no. 2, pp. 492–495, 2020.
- [5] Y. Ahn, S. J. Lee, and J. W. Shin, "Cross-corpus speech emotion recognition based on few-shot learning and domain adaptation," *IEEE Signal Processing Letters*, vol. 28, pp. 1190–1194, 2021.
- [6] J. Liu, W. Zheng, Y. Zong, L. U. Cheng, and C. Tang, "Cross-corpus speech emotion recognition based on deep domain-adaptive convolutional neural network," *IEICE Transactions on Information and Systems*, vol. E103.D, no. 2, pp. 459–463, 2020.
- [7] W. Zhang, P. Song, D. Chen, C. Sheng, and W. Zhang, "Cross-corpus speech emotion recognition based on joint transfer subspace learning and regression," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, pp. 588–598, 2021.
- [8] W. Zheng, W. Zheng, and Y. Zong, "Multi-scale discrepancy adversarial network for crosscorpus speech emotion recognition," *Virtual Reality & Intelligent Hardware*, vol. 3, no. 1, pp. 65–75, 2021.
- [9] L. Li and L. Cao, "Semantic analysis of literary vocabulary based on microsystem and computer aided deep research," *Mobile Information Systems*, vol. 2021, Article ID 8624147, 13 pages, 2021.
- [10] S. P. Yadav, S. Zaidi, A. Mishra, and V. Yadav, "Survey on machine learning in speech emotion recognition and vision systems using a recurrent neural network (RNN)," *Archives of Computational Methods in Engineering*, vol. 29, no. 3, pp. 1753–1770, 2022.
- [11] K. Chouhan, A. Shrivastava, C. Gangadhar, V. Shukla, and S. K. Jain, "Speech recognition classification with ANN implementation using machine learning algorithm," *Linguistica Antverpiensia*, vol. 2021, no. 1, pp. 2785–2796, 2021.
- [12] M. Rojc and I. Mlakar, "An LSTM-based model for the compression of acoustic inventories for corpus- based text-to-speech synthesis systems," *Computers and Electrical Engineering*, vol. 100, article 107942, 2022.
- [13] X. Ren, "Research on a software architecture of speech recognition and detection based on interactive reconstruction model," *International Journal of Speech Technology*, vol. 24, no. 1, pp. 87–95, 2021.
- [14] O. Ivanova, J. J. Meilán, F. Martínez-Sánchez, I. Martínez-Nicolás, T. E. Llorente, and N. C. González, "Discriminating speech traits of Alzheimer's disease assessed through a corpus

- of reading task for Spanish language,” *Computer Speech & Language*, vol. 73, article 101341, 2022.
- [15] I. Lefter, A. Baird, L. Stappen, and B. W. Schuller, “A cross-corpus speech-based analysis of escalating negative interactions,” *Frontiers in Computer Science*, vol. 4, article 749804, 2022.
  - [16] S. Kibria, A. M. Samin, M. H. Kobir, M. S. Rahman, M. R. Selim, and M. Z. Iqbal, “Bangladeshi Bangla speech corpus for automatic speech recognition research,” *Speech Communication*, vol. 136, pp. 84–97, 2022.
  - [17] A. Pandey and D. L. Wang, “Self-attending RNN for speech enhancement to improve cross-corpus generalization,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1374–1385, 2022.
  - [18] L. Xia, G. Chen, X. Xu, J. Cui, and Y. Gao, “Audiovisual speech recognition: a review and forecast,” *International Journal of Advanced Robotic Systems*, vol. 17, no. 6, 2020.
  - [19] C. M. Chen, M. C. Li, and M. F. Lin, “The effects of video-annotated learning and reviewing system with vocabulary learning mechanism on English listening comprehension and technology acceptance,” *Computer Assisted Language Learning*, vol. 35, pp. 1557–1593, 2020.
  - [20] R. Baumann, K. M. Malik, A. Javed, A. Ball, B. Kujawa, and H. Malik, “Voice spoofing detection corpus for single and multi-order audio replays,” *Computer Speech & Language*, vol. 65, article 101132, 2021.
  - [21] J. Basu, S. Khan, R. Roy, T. K. Basu, and S. Majumder, “Multilingual speech corpus in low-resource eastern and northeastern Indian languages for speaker and language identification,” *Signal Processing*, vol. 40, no. 10, pp. 4986–5013, 2021.
  - [22] J. Gideon, M. G. McInnis, and E. M. Provost, “Improving cross-corpus speech emotion recognition with adversarial discriminative domain generalization (ADDoG),” *IEEE Transactions on Affective Computing*, vol. 12, no. 4, pp. 1055–1068, 2021.
  - [23] A. Vempala and E. Blanco, “Extracting biographical spatial timelines: corpus and experiments,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1395–1403, 2020.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/360540906>

# Tick Tock Break The Clock: Breaking CAPTCHAs on the darkweb

Conference Paper · July 2022

DOI: 10.5220/0011273300003283

---

CITATIONS  
0

READS  
966

8 authors, including:



David Audran  
Aalborg University  
3 PUBLICATIONS 0 CITATIONS

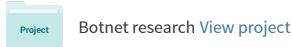
[SEE PROFILE](#)



Emmanouil Vasilomanolakis  
Technical University of Denmark  
66 PUBLICATIONS 986 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



# Tick Tock Break The Clock: Breaking CAPTCHAs on the darkweb

David Holm Audran, Marcus Braunschweig Andersen, Mark Højer Hansen, Mikkel Møller Andersen,  
Thomas B. Thomas, Kasper H. Hansen, Dimitrios Georgoulias and Emmanouil Vasilomanolakis

*Aalborg University, Copenhagen, Denmark*

{daudra21, mban21, mhha21, mman21, tfrede21, khha18}@student.aau.dk, {dge, emv}@es.aau.dk

**Keywords:** darkweb, CAPTCHA, web crawler, machine learning, darkweb marketplace, darkweb forum

**Abstract:** Nowadays, almost all major websites employ CAPTCHAs. This prevents website scraping, fake account creation as well as DDoS or brute-force attacks. For anonymity reasons, mainstream CAPTCHAs such as Google’s reCAPTCHA cannot be used on the darkweb. Due to the evolution of machine learning and computer vision, the CAPTCHA challenges used there, such as the clock CAPTCHA, are usually more arduous than those found on the clearweb. This paper presents an automated system that uses machine learning to break clock CAPTCHA challenges with a high success rate. We evaluate our system in a real world setting against 725 clock challenges from live darkweb marketplaces. Our results show an accuracy of 96.83% while maintaining low time requirements while analyzing, predicting and submitting the CAPTCHA solution.

## 1 INTRODUCTION

CAPTCHAs are widely used around the web to prevent an assortment of attacks such as DDoS, web crawling or creation of fake accounts. Since their inception in 1996 (Guerar et al., 2021) a long range of technologies have been developed. To attempt breaking these, a great deal of computer vision technologies have been utilized as the majority of CAPTCHAs are image-based. Accompanying this, machine learning can now be used as it greatly increases the computers’ chance to successfully solve such a puzzle.

The advancements of the aforementioned technologies, and especially of machine learning, have deprecated many CAPTCHA models. Despite the issues this may present, it also creates an invitation for creating new, more arduous models, that can distinguish a human from a machine. There is an ongoing arms race, where one side is trying to create resilient CAPTCHAs, while the other side is trying to break them. This is also the case on the darkweb.

The CAPTCHAs found on the darkweb differ from those on the clearweb since the desire to remain anonymous restricts which models can be used. Newer CAPTCHAs, such as reCAPTCHA, require a connection to Google services to check information about the client, and is therefore not a viable choice for darkweb sites. This has in turn increased the effort put into making different types of CAPTCHAs to be

utilized on the darkweb (Georgoulias et al., 2021).

Using machine learning to develop a model that can successfully solve CAPTCHA challenges requires a labelled data set containing examples of the CAPTCHA and corresponding answers. Such supervised machine learning approaches cannot be trained without proper labelled data. To fulfill this requirement the CAPTCHA could be downloaded and the answer manually filled in, however this could take a lot of time depending on the amount of data needed. Therefore having access to the source code of the CAPTCHA can automate its generation. However, in most cases, especially on the darkweb, the CAPTCHA code generation is proprietary and has to be reverse engineered. Furthermore when utilizing machine learning models, the model produced can be limited to solve only one type of CAPTCHA. A small modification to the CAPTCHA algorithm might render the machine learning model highly ineffective.

A predominant darkweb CAPTCHA scheme is the so-called clock CAPTCHA (Georgoulias et al., 2021) (see Figure 1a). The challenge is to correctly submit the time of an analogue clock that contains several misleading geometric shapes in under 60 seconds. The clock comes in different variations, however the general idea is the same. To this day, no method of breaking the clock scheme has been developed and disclosed. The main goal of this work is to break the basic version of the clock CAPTCHA

scheme along with one variation, utilizing machine learning. To achieve this goal we use the deep learning architecture ResNet50 to create a model which is able to predict the time of the clock CAPTCHA given an image of it. In addition, a web scraper is built providing the ability to automatically solve a challenge using the trained model in real time on a Tor hidden service. We limit ourselves and do not further expand our work on additional clock variations, since it would prove to be a never ending task, due to the highly dynamic nature of the darkweb.

One major challenge when attempting to automate data collection from platforms on the darkweb, is bypassing the Distributing Denial of Service (DDoS) protection mechanisms. Darkweb marketplaces and vendor shops utilize CAPTCHAs with the goal of taking away the automation capabilities of web crawlers (Soska and Christin, 2015). In essence, these mechanisms are put in place to force individuals into manually providing responses to challenges. This can be a time-consuming task, especially when attempting to deploy crawlers in multiple platforms simultaneously, making the entire process of data collection challenging. Successfully bypassing these mechanisms in an automated manner, provides ease and speed to the process, making research efforts more effective. Hence, we note that our work is only intended for assisting researchers and the developed system is available to researchers upon request, due to ethical considerations (see Section 2).

In this paper we show that: i) It is possible to develop a machine learning model to correctly solve the clock CAPTCHA with 96.83% accuracy; ii) the developed model can be utilized by a web scraper to access services using the clock CAPTCHA, in a timely manner; iii) the developed model can easily be adapted to incorporate modified versions of the clock while maintaining high accuracy.

## 2 ETHICAL ISSUES

In this section we want to address the ethical issues associated with this paper. Our work is not intended to be used in takedown attempts against the platforms implementing the showcased CAPTCHAs, since literature characterizes them as a non-violent alternative to street drug trafficking (Martin and Christin, 2016). Instead, our goal is to illustrate that solving the clock CAPTCHA can be automated, and then utilized by a web crawler to further automate data harvesting for research purposes.

With regard to the site access, the marketplaces that were part of our study are both publicly available

and free to access. Furthermore, we want to point out that we did not in any way hinder the operation of any of these platforms, or the experience of their users. In order to complete our experiments, we only used site mechanisms that are available to all users, and in a manner that did not consume any additional resources from the marketplaces' side. Lastly, our research did not involve any kind of user private data, hence there is no risk of exposing the identity or private information of any individuals.

## 3 BACKGROUND AND RELATED WORK

Even though CAPTCHAs on the clearweb and the darkweb have inherent differences, the methods for breaking CAPTCHAs are the same in both domains. Such methods, can be categorized in the following three categories: machine learning methods, non machine learning methods and hybrid methods.

### 3.1 Machine Learning Methods

The development of deep learning has resulted in great advances in CAPTCHA breaking. Challenges previously deemed impossible for computers to solve (e.g. advanced image recognition) are now solvable.

(Noury and Rezaei, 2020) showed a method for breaking text-based CAPTCHAs of a fixed length using convolutional neural networks, achieving up to 98.94% accuracy. Similarly, a deep learning method for breaking text-based CAPTCHAs are described by (Tang et al., 2018), where convolutional layers are combined with max-pooling layers.

When it comes to breaking image-based CAPTCHAs, sophisticated deep learning methods are necessary. Mittal et al. (Mittal et al., 2018) describe a method for breaking a CAPTCHA using the Inception V3 image recognition model, achieving a mean accuracy of 91% in real time.

Another example of using deep learning to break an image based CAPTCHA, is by (Hossen and Hei, 2021) using the neural-network ResNet18 architecture. The authors utilize a pre-trained instance of the model, that is trained using the ImageNet<sup>1</sup> data set. This minimizes the required training needed for the specific challenge. The focus of Hossen and Hei was to provide a low-cost method for breaking the CAPTCHA system, and they only required 143 minutes of training. They achieved an accuracy of 88% on the test set. Additionally, they achieved an accuracy

<sup>1</sup><https://www.image-net.org/>

of 95.93%, when providing the model with real-world examples of challenges.

## 3.2 Non Machine Learning Methods

While it is not widespread to apply non machine learning methods for breaking CAPTCHAs, there have been attempts using optical character recognition (OCR). (Csuka and Gaastra, 2018) propose a method using the OCR engine Tesseract (Smith, 2007) for breaking text-based CAPTCHAs on the darkweb. They describe the performance of this method as inferior to the applied machine learning method in terms of success rate, but it does operate faster and provides immediacy compared to the machine learning method.

According to (Weng et al., 2019) another non machine learning method for malicious activity, is using underground CAPTCHA solving services. These services consists of large amount of human labor solving CAPTCHAs in exchange for money.

## 3.3 Hybrid Methods

Any information the computer is able to retrieve or extract on the CAPTCHA challenge at hand improves the accuracy of the computers decision. A method used by (Sivakorn et al., 2016) aimed at breaking Google’s widely used reCAPTCHA (Shet, 2014) utilised both deep learning methods and Google’s own reverse image search engine. The challenge presented by reCAPTCHA is an image-based CAPTCHA, as described in Section 3.4 of this paper. The solution to breaking this, presented by (Sivakorn et al., 2016), consists of modules, that each assigns labels to an image. Most of the modules are deep learning based, but one of the modules is the Google reverse image search engine. The tags and labels provided by each module are then compared, and a decision for each image in the challenge is made.

## 3.4 Darkweb CAPTCHAs

CAPTCHA schemes like Google’s reCAPTCHA cannot be used on the darkweb, due to anonymity issues. For this reason, more traditional CAPTCHA scheme types are used. The most prominent ones being image-based CAPTCHAs and text-based CAPTCHAs (Georgoulias et al., 2021).

Text-based CAPTCHAs present an image of a string of random letters, with the goal being for the user to identify each letter. The images are often obscured, colored or blurred, to make it harder for

machines to recognize the letters, but still remaining fairly easy for a human.

Image-based CAPTCHAs work by presenting a question and a set of images to the user, and then requiring the user to pick an image/images thereof, that correspond to the question asked (Alqahtani and Alsulaiman, 2020). Alternatively, the user might be presented with a single image and are then required to answer the question by describing the image or its contents usually by picking from a set of options. The images shown are usually in poor quality and/or have shapes, lines or gibberish text that are easy for a person to distinguish from but hard for a machine. Image-based CAPTCHAs are considered to be the most advanced and secure type of CAPTCHA, as it is based on image details, which makes it hard for a machine to solve (Brodic et al., 2016).

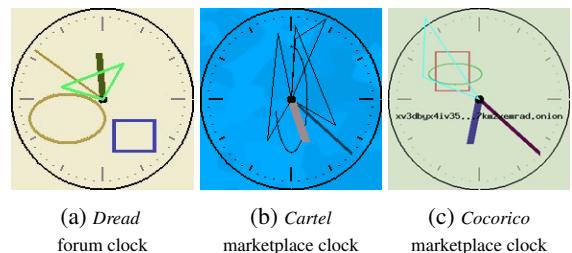


Figure 1: Three variations of the clock CAPTCHA in the darkweb

The CAPTCHAs found on the darkweb, although using these traditional ideas, have been improved upon, making them more innovative than the ones found on the clearweb. One of the most predominant schemes can be found on Figure 1, which illustrates different adaptations of the clock CAPTCHA. To solve the scheme, the correct time must be selected within a certain time frame. The challenge for machines being to distinguish the patterns and shapes from the clock’s pointers. This type of image-based CAPTCHA is widely adopted on darkweb marketplaces, with some variations depending on the hidden service. Two well-known hidden services the *Dread forum* and the *White House Market* have provided a public GitHub repository with code for what they call “EndGame V2 - Onion Service DDOS Prevention Front System”<sup>2</sup>. This system provides several services with one of them being the clock CAPTCHA scheme as seen on Figure 1a. Due to this variation being publicly available, it is one of the more commonly used. Other variations can be seen on Figures 1b and 1c. The clock found on the *Cartel* marketplace differentiates by having a different background with patterns as well as placing shapes around the center.

<sup>2</sup><https://github.com/onionltd/EndGame>

The *Cocorico* marketplace, places the url of the site horizontally across the middle of the clock, which is a technique often used on darkweb CAPTCHAs.

## 4 THREAT MODEL

In this section we discuss the threat model followed in this paper in terms of the capabilities of the attacker and the required accuracy of an attack to be considered practically successful.

### 4.1 Attacker capabilities

We assume that the attacker is able to produce labelled data for the supervised machine learning algorithm. This assumption comes with the limitation that without labelled data the whole process would require significant manual work.

Moreover, we assume that the attacker is able to both contact the (Tor) hidden service of the target website and is also able to download an image of the CAPTCHA clock. Furthermore, on the one hand the attacker requires high computational power (especially RAM) to be able to train the residual neural network that we will be utilizing in this paper. On the other hand, upon training the model, the attacker is able to run the prediction on a typical computer with no significant computational capabilities. In that sense, and excluding the training phase, our threat model follows the work of (Bock et al., 2017).

### 4.2 Attack accuracy

The definition of when a CAPTCHA scheme is considered broken is not simple. The debate for this is very opinionated and no definitive accuracy threshold has been agreed upon (Bursztein and Bethard, 2009).

When designing a CAPTCHA scheme the original design goal states that “*automatic scripts should not be more successful than 1 in 10000 attempts*”, which equates to an accuracy of 0.01% (Chellapilla et al., 2005). This is widely regarded as too ambitious, as random guesses would be able to reach an accuracy higher than this. Instead, many regard 1% accuracy to be the threshold, as the accuracy of random guesses would be within the acceptable margin, and therefore not able to deem the scheme broken (Bursztein et al., 2011). Others argue that 5%, or even higher percentages, are more reasonable (Baecher et al., 2010).

For attackers aiming at breaking CAPTCHAs, the accuracy goal is usually a lot higher. (Hossen and Hei, 2021) present an accuracy goal of above 50%, aiming at developing a low-cost attack against the

hCaptcha system. (Aboufadel et al., 2005) state that a CAPTCHA is considered broken if a computer algorithm can solve the scheme 4 out of 5 times on average, implying an accuracy goal of above 80%.

In reality, the viable accuracy is dependent on the amount of resources the attacker possesses and the cost of the attack (Bock et al., 2017). An attacker with many resources, that would be able to attack the CAPTCHA scheme tens or hundreds of thousands times, might only need an accuracy of 1% for the attack to be worthwhile. Similarly, an attacker with limited resources might need an accuracy of above 50% to even consider the attack. Furthermore, many darkweb sites implement a lockout function, that blacklists the user if the CAPTCHA scheme has been failed three times. This obstacle demands a certain level of accuracy of the attack, for it to be viable to use in automation, i.e., with a web scraper.

Based on the aforementioned previous work and the fact that in most darkweb marketplaces and forums one can try to solve a CAPTCHA at least twice before being blacklisted we expect an accuracy higher than 80% to be more than satisfactory. For a 80% model success rate and the user having 2 attempts to successfully solve the CAPTCHA, the probability  $P$  of a crawler providing the correct solution at least once is calculated at 96% :

$$\begin{aligned} P(\text{SucceedAtLeastOnce}) &= 1 - P(\text{FailBothAttempts}) \\ &= 1 - 1^{\text{st}}\text{Fail} * 2^{\text{nd}}\text{Fail} \\ &= 1 - 20\% * 20\% \\ &= 96\% \end{aligned}$$

## 5 SYSTEM OVERVIEW

Our automated CAPTCHA breaking system solves the darkweb clock scheme using machine learning. The system can be reduced into three main steps: i) model setup, ii) model training and iii) model usage. 2. The first two steps take place on an AI cloud, where

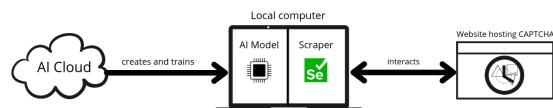


Figure 2: Architectural overview of the entire CAPTCHA breaking system

the model is set up and trained with the generated pictures. The AI Cloud is a separate system, which is utilized due to the computational resources it provides. The model is then downloaded and saved to a local computer. On the local computer the model is used by the scraper, which connects to a Tor site, that uses

the clock CAPTCHA. The scraper downloads the image from the CAPTCHA and predicts the answer to it, using the model from the AI Cloud. The architectural overview of the system is visualized in Figure The system depends on two different programming environments. The image generation is C++ code, based on the Lua code found in the EndGame repository, and the machine learning training and web scraper is created in Python. The aforementioned individual steps are elaborated in the following sections.

## 6 RESNET50 VS THE DARKWEB CLOCK CAPTCHA

In this section we go over the details that surround setting up, training, and using the model.

### 6.1 Setting up the model

We divide the model assembly into 3 distinct procedures: the data generation, preprocessing, and the use of ResNet50.

#### 6.1.1 Data generation

In order to obtain labeled data, the code for the clock CAPTCHA was extracted from the EndGame code base mentioned in Section 3.4 and rewritten into a C++ program with additional functionality. The program is able to generate PNG clock images along with a text file containing the time shown on each of the generated clocks. The program has the ability to generate both the clock from the code base, and the clock found on the Cartel marketplace, as seen on Figure 1b. The program takes two arguments, where the first argument is the number of iterations. One iteration will generate 720 images for each of the two clock types. Hereof the number 720 stems from the fact that an analogue clock is divided into 12 hours and 60 minutes, amounting to  $12 * 60 = 720$  different clock hand positions. The program will generate an image for each of these possibilities. The second argument should be either 0, 1 or 2, and will determine if the program should generate the Dread forum clock (0), the Cartel marketplace clock (1), or both (2).

#### 6.1.2 Preprocessing

In order for the model to be able to train on the generated data, it is first necessary to preprocess the images. First, the images are loaded along with the corresponding labels. We decided to retain as much information as possible from the images, and therefore

kept all 3 RGB channels in the picture instead of converting them to e.g., greyscale. The size of the image is then re-scaled to ensure that it is 190x190 pixels, as that is the size used in most cases in the darkweb. The images are then normalized, changing the pixel intensity range values from 0 – 255 to 0 – 1. This makes it easier for the model to converge, and limits the amount of zero-gradients during training.

#### 6.1.3 Using ResNet50

To solve the clock CAPTCHA we use a residual neural network. The challenge lies withing accurately determining the time on the clock, which has 720 possible solutions. Therefore, the type of deep learning problem is a multi-class classification problem with 720 classes, with one class per possible time on the clock. Provided a CAPTCHA challenge, the classifier generates 720 probabilities, each giving how likely the corresponding class is for the provided challenge. With this list of probabilities, it is then possible to extract the highest one to find the classifiers best guess for a solution to the challenge.

The architecture we have chosen to use is the ResNet50 architecture. This architecture consists of an initial convolutional layer followed by a max pooling layer, 48 convolutional layers divided into residual building blocks with 3 layers in each, aw well as an average pooling layer. Moreover, we added a dropout layer with a rate of 0.7, that randomly sets inputs to 0, with a frequency of the rate at each step<sup>3</sup>. This aids in avoiding over-fitting the model, essentially dropping some information randomly during training. In addition, a final dense layer is added, mapping the output of the final layer to the number of possible classes, in our case 720, using softmax activation.

The entire implementation of our deep learning model is implemented in Python using the Keras API <sup>4</sup>. Keras provides both an untrained and trained version of the ResNet50 architecture. We utilized the untrained architecture and performed the necessary alterations of it to suit our challenge.

### 6.2 Training the model

Training of a deep residual network is a complex and resource heavy task, which can take a long time. The model used in this paper builds upon the ResNet architecture, and is trained on an AI Cloud with optimized hardware for this specific task. The node which

---

<sup>3</sup>[https://keras.io/api/layers/regularization\\_layers/dropout/](https://keras.io/api/layers/regularization_layers/dropout/)

<sup>4</sup><https://keras.io/>

training has been performed on contains 96 'Intel(R) Xeon(R) Platinum 8168 CPU @2.70GHZ' CPUs, has 128GB RAM allocated and utilizes two 'Tesla V100-SXM3-32GB' GPUs to parallelize the training. Being able to utilize hardware this powerful cut the otherwise long training phase very short.

While training the deep learning model, we found that a batch size of 64 gave the best results. In order to optimally utilize the GPUs available to us for training, the batch size is scaled up according to the number of GPUs, giving each GPU the intended batch size to work with. The same is done for the learning rate.

The dataset used consists of 72,000 samples, and is split into 80% for training and 20% for testing. The training set is split up further, using 20% as the validation set. The reasoning behind this, is with access to the publicly available source code used for generating the CAPTCHA challenge we are attacking, we solved the challenge of data collection by being able to generate our own labelled data automatically. This allowed for the generation of perfectly balanced datasets of any size, that are identical to the data the model would be faced with in the evaluation and testing phase. Furthermore the whole dataset is shuffled before being split up into training and test sets, to ensure that every class is represented in each of the sets.

The metrics the model is judged by is the *loss* and the *accuracy*. The *loss* is sparse categorical cross entropy loss, a function used to calculate the loss of the predictions made by the model in its current state, compared to the true labels. The loss is used to indicate to the model, how well it predicted in the current iteration of training. The categorical cross entropy for  $n$  number of predictions, is defined as:

$$Loss = - \sum_{i=1}^n y_i \cdot \log \hat{y}_i \quad (1)$$

where  $y_i$  is the actual label, and  $\hat{y}_i$  is the prediction made by the model. The *accuracy* is standard classification accuracy, i.e., the number of accurate predictions divided by the total number of predictions. The reasoning behind using standard classification accuracy is that the dataset which the model builds upon is equally distributed among all possible classifications.

Three different models have been trained on the AI Cloud. Initially our focus was the model with the most generic type of the clock, as seen on Figure 1a. The first edition of the model was only trained on the Endgame variation of the clock. The results from the Dread forum clock showed a high accuracy and low loss (see Table 1). However, when tested against the clock variations illustrated on Figures 1b and 1c, the model was unsuccessful. At that point it was obvious that the slightest changes in the clock resulted in the

performance of the model deteriorating drastically. To combat this, but to also test the adaptability of the model training approach to modified versions of the clock, clocks like the one found on the Cartel marketplace had to be generated. After analyzing the properties of the specific clock variation, and modifying the clock generation code, we were able to successfully generate these as well. Hence, we decided to build a combined model for two clock types which is trained on a equally distributed amount of both clocks at a 1:1 ratio, 36,000 samples of each. The training was set to 200 epochs, with early stopping if the validation loss reaches a lower value than 0.05, using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001. Training on the generic clock alone lasted for 708 seconds, reaching 11 epochs. Training on the Cartel marketplace clock lasted for 301 seconds, reaching 4 epochs. Training on the combined model lasted for 759 seconds, reaching 12 epochs. The results of the training can be seen in Table 1.

Model	Accuracy	Loss
Dread Forum Clock	0.992	0.029
Cartel Marketplace Clock	0.996	0.025
Combined	0.988	0.048

Table 1: Overview of the performance of the different models on the test set.

### 6.3 Using the model

To use the model a web scraper was developed capable of navigating to a given list of URLs either via direct input or via reading the URLs from a text file. The scraper loads the machine learning model and utilizes Selenium<sup>5</sup> to open a web browser and navigate to the site. The browser chosen in this case was Google Chrome which does not support Tor natively however does support utilizing a proxy to connect with. A Tor proxy was therefore set up on the default 9050 port. After connecting the webdriver which selenium operates by, the browser navigates to the site requested. As most of these sites contain a queuing system to avoid DDoS attacks, it waits until the clock CAPTCHA appears on screen before continuing. Once this happens the scraper finds the clock image, downloads it, resizes it to 190x190 pixels, passes it into the model and awaits its response. Finally, the scraper passes the response to the site and clicks the submit button. If the prediction is accurate, the CAPTCHA is bypassed successfully. If the model produces an incorrect result, the site generates a new clock challenge and the procedure is repeated. Should this result also be incorrect, the scraper exits

<sup>5</sup><https://www.selenium.dev/>

the site, since after the third incorrect submission the Tor identity of the scraper will be banned from the site. The scraper automatically detects whether or not it was successful by checking if the CAPTCHA exists after clicking the submit button. If not, it assumes it has successfully bypassed the CAPTCHA mechanism and navigated to the home page of the site.

The trained model, along with its weights is loaded by the scraper using the Keras API. A function in the Python script used to make predictions, is then able to make a prediction for a single image, and return the numerical label as a tuple, in an hour/minute format. It then utilizes a dictionary loaded from a JSON file, containing the mapping of numerical labels to actual labels, to convert the values.

To test the model we utilize the scraper on 9 popular darkweb websites, that either contain the Cartel marketplace or the Dread Forum CAPTCHA clock variation. Each website is visited 20 times, however visits that experience connection errors are being excluded from the final data set. The time measurements are taken purely on the runtime of each metric, while ignoring any time used on waiting in the DDoS queue, connecting to the site, etc.

## 7 RESULTS AND DISCUSSION

To perform the evaluation of our deep learning model we will be using the SKLearn Metrics module<sup>6</sup>, which provides a function to write out a classification report. It provides the metrics precision, recall and F1-score for each class in the data set, and an average for each metric across all classes. For each class,

	Precision	Recall	F1-score	Support
0:0	1.00	1.00	1.00	20
0:1	1.00	1.00	1.00	20
0:2	1.00	1.00	1.00	20
...				
...				
11:58	1.00	0.90	0.95	20
11:59	1.00	1.00	1.00	20
Accuracy			0.99	14400
Macro avg	0.99	0.99	0.99	14400
Weighted avg	0.99	0.99	0.99	14400

Table 2: ResNet50 model performance test set

a true positive is correctly labelling the image as the given class, and a false positive is labelling the image as the given class even though it is not. A true negative is correctly not labelling the image as the given class, and a false negative is incorrectly not labelling

<sup>6</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification\\_report.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html)

the image as the given class.

The evaluation of our model will be performed on a new labelled test data set consisting of 14,400 images, with a combination of two different variations of the clock CAPTCHA. This test data set was balanced with 10 instances of each possible class for both variations of the clocks. The evaluation was performed on a standard computer with an Intel i5-4210U (1.70 GHz) CPU and 8GB of RAM, and the Ubuntu 20.04.3 LTS operating system.

The time required for this computer to load the model, load all of the 14,400 challenges and provide a solution was 41 minutes and 44 seconds. This is an average of 0.17 seconds for each prediction on a standard computer. As shown in Table 2, our model achieves an accuracy of 99% on the test data set, and an average precision, recall and F1-score of 99% across all classes.

### 7.1 Clocks in the wild

The scraper was programmed with the functionality to run in both a sequential and a parallel mode. The parallel mode utilizes a thread-pool allowing the scraper to run on several sites at once. Nevertheless, due to the fact that threads share resources, the time to solve a CAPTCHA on a site is increased quite drastically.

#### 7.1.1 Sequential Mode

In the sequential mode, the scraper performed very well in both finding the CAPTCHA and inserting the result as indicated in Figure 3. The scraper utilized the *combined* model as described in Section 6.2 as the initial *Dread forum* model was unable to bypass the clock found on the Cartel marketplace (see Figure 1b). As seen in Figure 3, the time spent by the scraper to find and download the image is about 0.05 seconds on average. The prediction itself averages at 0.12 seconds, and inserting the answer that the model predicted takes 0.3 seconds, all in all resulting in a scraper which can solve a clock CAPTCHA scheme on a site in approximately 0.5 seconds on average.

#### 7.1.2 Parallel Mode

In parallel mode, discovering the CAPTCHA image and acquiring it took an average of 0.6 seconds, while the prediction from the model averaged at 2.6 seconds. Lastly, the result submission was executed in approximately 3.6 seconds, contributing towards an average total of 6.9 seconds, for the entire process (see Figure 4). The parallel mode does also require more computational resources, since Selenium opens a new browser instance for each URL to scrape, while

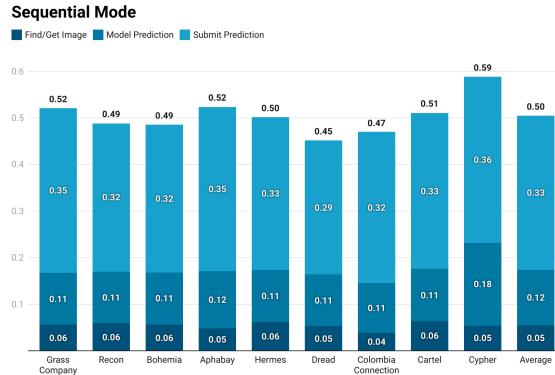


Figure 3: Average runtime of the scraper in seconds, per site and in sequential mode. The presented data were calculated only from successful connections.

the sequential mode opens a new tab in the same browser instance. Hence, choosing the optimal mode depends on the use case (e.g. commencing crawling on one platform at a time, or several at the same time).

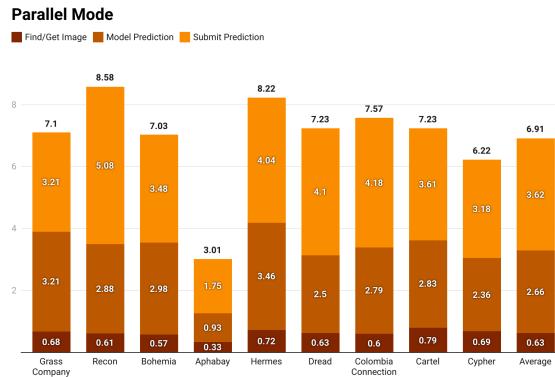


Figure 4: Average runtime of the scraper in seconds, per site and in parallel mode. The presented data were calculated only from successful connections.

### 7.1.3 Mode Comparison

Running the scraper in sequential mode on all 9 marketplaces is calculated at a total average of 4.5 seconds, for all of the CAPTCHA challenges to be successfully solved. In parallel mode, this number goes up to 6.9 seconds and is equal to the average presented on Figure 4, since in the specific mode this number is already calculated with all of the CAPTCHAs being solved simultaneously. We consider these two averages to be important since they provide an estimation of the time a user would need to run the scraper one platform at a time or on several platforms concurrently, without the identity of the platforms being a factor. Furthermore, the disparity between the two results, is attributed to the aforementioned need for ad-

ditional computational resources that the scraper requires when operating in parallel mode.

### 7.1.4 Scraper Evaluation

We tested our system by performing a total of 702 marketplace visits, both in parallel and sequential mode. As shown on Table 3, the scraper had to perform an extra attempt to solve the challenge in 23 occasions. This translates into the scraper being able to solve the CAPTCHAs from the 679 remaining marketplace visits, on the first try. The resulting number of challenges solved amounts to a total of 725, with an overall accuracy of 96.83% (702 out of 725). Lastly, in all of the 702 visits, regardless of whether it took one or two attempts, the scraper managed to provide automated access to the platform via bypassing the CAPTCHA mechanism in 100% of the cases.

The results described above were achieved on a computer with an Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz and 8GB of RAM running the Parrot Linux operating system version 5.0.

CAPTCHAs	Retries	Accuracy	Site Visits	Overall success
725	23	96.83%	702	702 / 702 (100%)

Table 3: Performance of the scraper

## 8 CONCLUSION

In this work, we present a high performance attack on the clock CAPTCHA found on multiple darkweb marketplaces and forums, utilizing a deep residual machine learning model, trained with a self generated dataset, on a high performance AI Cloud. The result is a model achieving an F1-score of 0.99 on 14,400 separately generated clock instances. Combining this model with a web scraper, we successfully tested our system against 725 CAPTCHA challenges, which belong to two different variations of the darkweb clock CAPTCHA, with a 96.83% accuracy.

One limitation of this paper, is that our model is over-fitted, fitting too closely to the training set and thus does not perform well on unseen data. The cause of this issue stems from the fact that the dataset is uniform in terms of the clock image itself. The model places great importance on the features of these specific clocks, hence modifying the target CAPTCHAs results in a weaker performance of the model. However, we do illustrate that adapting the training data to different variations of the clock, which we can easily generate by modifying our data generation program,

is an effective solution to the over-fitting problem. This gives our automated CAPTCHA solving system great adaptability for future changes.

Another limitation is that the training of a model requires a lot of memory. Training on 72,000 images requires somewhere between 64 – 128GB of RAM, which is not available on a standard computer. This requirement of RAM stems from the individual image file size including all RGB channels and the sheer amount of images we used to train the model.

Lastly, with the goal of further improving our current system, we also experimented with the Resnet18 architecture. We trained a new model using the exact same parameters as we did with the Resnet50 architecture and evaluated it with the SKLearn Metrics module. Our preliminary results suggest that the model is able to achieve an accuracy of 100%, with an average precision, recall and F1-score of 100% across all classes, showing a lot of promise for future implementations. Since Resnet18 is a significantly lighter architecture than Resnet50, we will focus on this model in our future work.

## REFERENCES

- Aboufadel, E., Olsen, J., and Windle, J. (2005). Breaking the holiday inn priority club captcha. *The College Mathematics Journal*, 36(2):101–108.
- Alqahtani, F. H. and Alsulaiman, F. A. (2020). Is image-based captcha secure against attacks based on machine learning? an experimental study. *Computers & Security*, 88:101635.
- Baecher, P., Fischlin, M. G. L., Langenberg, R., Lützow, M., and Schröder, D. (2010). Captchas: The good, the bad, and the ugly. *Sicherheit 2010. Sicherheit, Schutz und Zuverlässigkeit*.
- Bock, K., Patel, D., Hughey, G., and Levin, D. (2017). uncaptcha: a low-resource defeat of recaptcha’s audio challenge. In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*.
- Brodić, D., Petrovska, S., Jevtić, M., and Milivojević, Z. N. (2016). The influence of the captcha types to its solving times. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1274–1277.
- Bursztein, E. and Bethard, S. (2009). Decaptcha: breaking 75% of ebay audio captchas. In *Proceedings of the 3rd USENIX conference on Offensive technologies*, volume 1, page 8. USENIX Association.
- Bursztein, E., Martin, M., and Mitchell, J. (2011). Text-based captcha strengths and weaknesses. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 125–138.
- Chellapilla, K., Larson, K., Simard, P. Y., and Czerwinski, M. (2005). Building segmentation based human-friendly human interaction proofs (hips). In *International Workshop on Human Interactive Proofs*, pages 1–26. Springer.
- Csuka, K. and Gaastra, D. (2018). Breaking captchas on the dark web.
- Georgoulias, D., Pedersen, J. M., Falch, M., and Vasilemanolakis, E. (2021). A qualitative mapping of dark-web marketplaces. In *Symposium on Electronic Crime Research (eCrime)*. IEEE.
- Guerar, M., Verderame, L., Migliardi, M., Palmieri, F., and Merlo, A. (2021). Gotta captcha ‘em all: A survey of twenty years of the human-or-computer dilemma. 2021-10-06.
- Hossen, M. I. and Hei, X. (2021). A low-cost attack against the hcaptcha system.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Martin, J. and Christin, N. (2016). Ethics in cryptomarket research. *International Journal of Drug Policy*, 35:84–91.
- Mittal, S., Kaushik, P., Hashmi, S., and Kumar, K. (2018). Robust real time breaking of image captchas using inception v3 model. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pages 1–5.
- Noury, Z. and Rezaei, M. (2020). Deep-captcha: a deep learning based CAPTCHA solver for vulnerability assessment. *CoRR*, abs/2006.08296.
- Shet, V. (2014). Are you a robot? introducing “no captcha recaptcha”. <https://security.googleblog.com/2014/12/are-you-robot-introducing-no-captcha.html>.
- Sivakorn, S., Polakis, J., and Keromytis, A. D. (2016). I’m not a human : Breaking the google recaptcha.
- Smith, R. (2007). An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633.
- Soska, K. and Christin, N. (2015). Measuring the longitudinal evolution of the online anonymous marketplace ecosystem. In *24th USENIX security symposium (USENIX security 15)*, pages 33–48.
- Tang, M., Gao, H., Zhang, Y., Liu, Y., Zhang, P., and Wang, P. (2018). Research on deep learning techniques in breaking text-based captchas and designing image-based captcha. *IEEE Transactions on Information Forensics and Security*, 13(10):2522–2537.
- Weng, H., Zhao, B., Ji, S., Chen, J., Wang, T., He, Q., and Beyah, R. (2019). Towards understanding the security of modern image captchas and underground captcha-solving services. *Big Data Mining and Analytics*, 2(2):118–144.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/367462058>

# A Systematic Review of Green AI

Preprint · January 2023

---

CITATIONS

0

READS

81

3 authors:



**Roberto Verdecchia**  
University of Florence (UniFI)

44 PUBLICATIONS 402 CITATIONS

[SEE PROFILE](#)



**June Sallou**  
Université de Rennes 1

8 PUBLICATIONS 33 CITATIONS

[SEE PROFILE](#)



**Luís Cruz**  
Delft University of Technology

50 PUBLICATIONS 492 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



PhD project on using modelling for decision making and risk assessment in hydrological information systems. [View project](#)

# A Systematic Review of Green AI

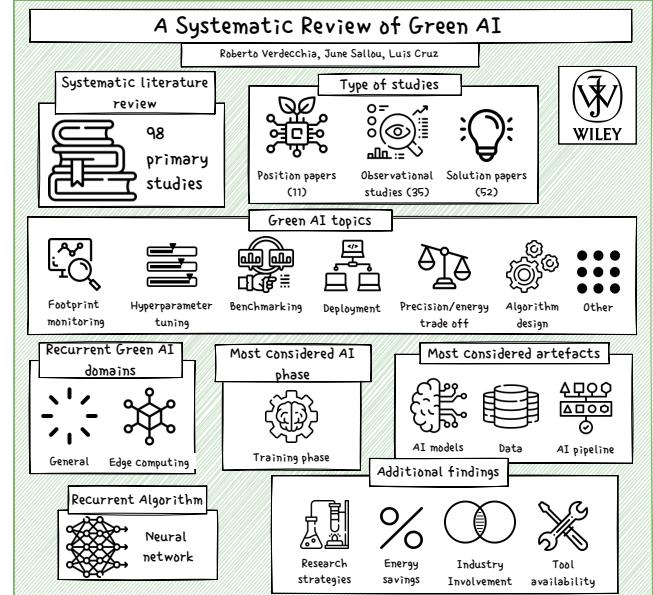
Roberto Verdecchia  
Vrije Universiteit Amsterdam  
Amsterdam, The Netherlands  
r.verdecchia@vu.nl

June Sallou  
TU Delft  
Delft, The Netherlands  
j.sallou@tudelft.nl

Luís Cruz  
TU Delft  
Delft, The Netherlands  
l.cruz@tudelft.nl

## Abstract

With the ever-growing adoption of AI-based systems, the carbon footprint of AI is no longer negligible. AI researchers and practitioners are therefore urged to hold themselves accountable for the carbon emissions of the AI models they design and use. This led in recent years to the appearance of researches tackling AI environmental sustainability, a field referred to as Green AI. Despite the rapid growth of interest in the topic, a comprehensive overview of Green AI research is to date still missing. To address this gap, in this paper, we present a systematic review of the Green AI literature. From the analysis of 98 primary studies, different patterns emerge. The topic experienced a considerable growth from 2020 onward. Most studies consider monitoring AI model footprint, tuning hyperparameters to improve model sustainability, or benchmarking models. A mix of position papers, observational studies, and solution papers are present. Most papers focus on the training phase, are algorithm-agnostic or study neural networks, and use image data. Laboratory experiments are the most common research strategy. Reported Green AI energy savings go up to 115%, with savings over 50% being rather common. Industrial parties are involved in Green AI studies, albeit most target academic readers. Green AI tool provisioning is scarce. As a conclusion, the Green AI research field results to have reached a considerable level of maturity. Therefore, from this review emerges that the time is suitable to adopt other Green AI research strategies, and port the numerous promising academic results to industrial practice.



**Graphical Abstract: From a systematic review of the Green AI literature, Green AI results to focus on solutions, and is often not bound to a specific context or algorithm. The Green AI research field results to be mature, i.e., the moment is suitable to port results from academic research to industrial practice.**

## 1 Introduction

In recent years, the Artificial Intelligence (AI) community has been challenged to bring the carbon footprint of AI models to the top of their research agenda. The iconic paper by Strubell et al. [93] analyzes the carbon impact of training their own state-of-the-art models. Results lead to the conclusion that we need to reduce the carbon footprint of developing and running AI models.

This self-reflection was an eye-opener to the AI research community. Many papers followed, calling for a new research direction that would consider this problem. Schwartz et al. coined the term Green AI as “*AI research that yields novel results while taking into account the computational cost*” [88]. Bender et al. published a position paper highlighting the consequences of continuously increasing the size of AI models [27]. A natural question that is posed is whether we are doing enough as a research community to mitigate the carbon impact of developing and running AI-based software.

AI systems are significantly complex and, to achieve Green AI, we need a joint effort that targets all the different stages of an AI

system’s lifecycle (e.g., data collection, training, monitoring), different artifacts (e.g., data, model, pipeline, architecture, hardware), etc [3].

Given the heterogeneity of the field, it is also difficult to have a broad view of all the Green AI literature that has been published in the past years. To understand the existing research, we conduct a systematic literature review on Green AI. We provide an overview and characterization of the existing research in this field. Moreover, we study how the field has been evolving over the years, pinpoint the main topics, approaches, artifacts, and so on.

This literature review shows that there has been a significant growth in Green AI publications – 76% of the papers have been published since 2020. The most popular topics revolve around monitoring, hyperparameter tuning, deployment, and model benchmarking. We also highlight other emerging topics that might lead to interesting solutions – namely, *Data Centric Green AI*, *Precision/Energy Trade-off* analysis. The current body of research has already showcased promising results with energy savings from 13% up to 115%. Still, most of the existing work focuses on the training stage of the AI model. Moreover, we observe that there is little involvement

of the industry (23%) and that most studies revolve around laboratory experiments. We argue that the field is growing to a level of maturity in which involvement of the industry is quintessential to enable the overarching goal of Green AI: harness the full potential of AI without a negative impact in our planet.

To encourage open science and the reproducibility of this study, we provide all data and scripts in a replication package available online with an open source license<sup>1</sup>.

The remainder of this paper is structured as follows. In Section 2, we describe the methodology used to collect and analyze Green AI literature. In Section 3, we present all the results yielded by our methodology. Section 4 discusses findings and reflects on the impact of our results in the research community. In section 5, we reflect on the potential threats of the validity of this study. Following, Section 6 describes related work and pinpoints the differences with our study. The main conclusions and future work are presented in Section 7.

## 2 Methodology

In this section, we document the research design, which was rigorously adhered to during the planning and execution of the study. We primarily followed the guidelines for conducting SLRs in software engineering research presented by Kitchenham [6].

### 2.1 Research Objective and Question

The goal of this review is to understand the characteristics of existing Green AI research. By utilizing the Goal-Question-Metric method [1], this objective can be described more formally as follows:

*Analyze Green AI literature*

*For the purpose of knowledge collection and categorization*

*With respect to AI*

*From the viewpoint of researchers and practitioners*

*In the context of environmental sustainability.*

The goal of this research can be directly translated in a research question (RQ), which states as follows:

**RQ1:** *What are the characteristics of Green AI state-of-the-art research?*

By answering our research question, we aim at gaining a systematic overview of the Green AI body of knowledge, starting from an outline of the general publication trends, to a detailed analysis of the past and current Green AI research activities and their characteristics.

### 2.2 Research Process

An overview of the research process followed is depicted in Figure 1. The process starts with the execution of a conservative automated search query via the digital libraries and indexing platforms *Google Scholar*, *Scopus*, and *Web of Science*, complemented by a subsequent iterative bidirectional snowballing process, which is conducted until the achievement of theoretical saturation. Including multiple literature indexing platforms to execute the automated search allows us to conduct an encompassing search of the literature based on multiple sources, hence allowing us to mitigate potential threats

<sup>1</sup>Replication package: <https://github.com/luiscruz/slrb-green-ai>

to external validity, as further documented in Section 5. Following, the details of each step of our research process are documented in detail.

**2.2.1 Automated Initial Search.** To identify a preliminary set of potentially relevant research works, we design an encompassing automated query to be executed on three different literature indexes, namely *Google Scholar*, *Scopus*, and *Web of Science*. The automated query targeting publication titles states as follows:

#### Listing 1: Automated search query

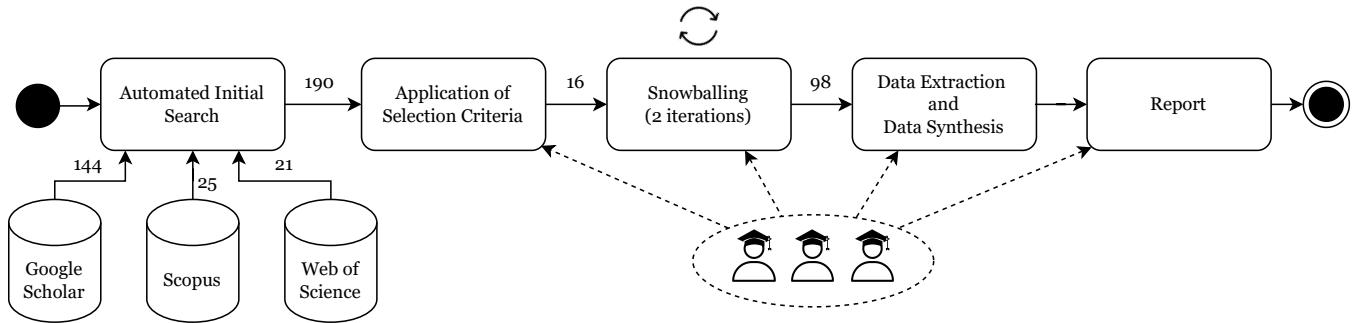
```
1 INTITLE("green" OR "sustainab*") AND
2 INTITLE("AI" OR "ML" OR "artificial_intelligence"
3 OR "machine_learning" OR "deep_learning")
```

The query is designed to retrieve literature with titles containing keywords related to sustainability, identified by the keywords *green* or *sustainability* and its variations, e.g., “*sustainable*” (Listing 1, Lines 1). The second part of the query instead is used to retrieve literature concerning AI, or related synonyms and acronyms (Listing 1, Lines 2-3). The query is executed on the three aforementioned literature libraries and indexes on the 18th of July 2022, and led to the identification of 190 potentially relevant studies. In order to be as comprehensive as possible, and avoid potential threats to external validity, the year of publication is left unbounded in the automated search.

**2.2.2 Application of Selection Criteria.** Subsequent to the identification of the initial potentially relevant studies, we execute the manual selection of the studies via a set of selection criteria defined *a priori*. A paper is confirmed as primary study if it adheres to all inclusion criteria, and none of the exclusion ones. The following inclusion (I) and exclusion (E) criteria are used:

- I1- The study regards AI
- I2- The study regards environmental sustainability
- I3- The study regards the environmental sustainability of AI
- I4- The study regards the software level
- E1- The study is not written in English
- E2- The study is not available
- E3- The study is a duplicate or extensions of an already included study
- E4- The study is a secondary or tertiary study
- E5- The study is in the form of editorials, tutorials, books, extended abstracts, etc.
- E6- The study is a non-scientific publication or grey literature

With the first three inclusion criteria (I1-I3), we ensure that the primary studies focus on Green AI (I1, I2), and that the studies regard the environmental sustainability of AI, rather than the improvement of environmental sustainability *through* AI. With the fourth inclusion criterion instead (I4), we ensure that the primary studies focus on software-centric Green AI. This latter criterion is used to exclude studies focusing on hardware-specific Green AI techniques, e.g., the use of *ad hoc* implemented hardware components, which we consider out of reach for most researchers/practitioners interested in Green AI, and is only marginal to the definition of Green AI itself [88].



**Figure 1: Systematic literature review process overview.**

The exclusion criteria are designed to ensure that data can be extracted from the papers (E1, E2), do not represent duplication or redundancy with respect to other primary studies (E3, E4), and are provided in the form of scientific studies (E5, E6).

To ease the primary study selection process, adaptive reading depth [11] is used to efficiently assess potentially relevant studies. In order to mitigate subjective biases and interpretations, the three authors independently utilized the selection criteria to scrutinize 63–64 candidate studies. Weekly meeting are held during the selection process to jointly discuss examples, doubts, and align the selection process between the three researchers.

The application of the selection criteria concludes with the identification of 16 primary studies, which constitute the starting set for the subsequent snowballing process.

**2.2.3 Snowballing.** In order to enrich the set of selected primary studies, and ensure that the primary study comprehensively represents the Green AI body of literature, the automated search results are complemented with a recursive bidirectional snowballing process [17]. This step entails the scrutiny of all studies either citing or cited by the already included primary studies. As for the application of selection criteria, three researchers are involved in the snowballing. During each snowballing round, the researchers independently snowball different primary studies, and propose new primary studies to be included, *i.e.*, the new identified studies which adhere to the selection criteria. During each snowballing round, examples, doubts, and divergences are jointly revisited and resolved, and the next snowballing iteration is started. A total of two rounds of backward and forward snowballing are executed before no new studies are identified, *i.e.*, when theoretical saturation is reached. The snowballing process terminates with the inclusion of 82 new primaries studies, leading to a total of 98 primary studies which are considered in the literature review reported in this research.

**2.2.4 Data Extraction.** In order to achieve the intended goal of this study and answer our RQ (see Section 2.1), we proceed to systematically extract data from the primary studies. The data extraction process consisted of two subsequent phases.

The first phase consists of a data exploration process, which terminates with the establishment of the data extraction framework of this study. Specifically, during this first phase, the three authors of this review independently scan the identified primary studies, and annotate the characteristics of the studies which are relevant

to answer our RQ. The identified characteristics are then jointly discussed and refined, leading to the consolidation of the fields constituting the data extraction framework of this review.

In the second data extraction phase, the primary studies are thoroughly analyzed, and the data is extracted from the studies according to the data extraction framework.

The fields of the data extraction framework utilized for this literature review on Green AI are the following.

- *Study type:* The overarching type of study, which could be either presenting a position on Green AI, a Green AI solution, or an observational study on Green AI;
- *Topic:* The Green AI topic considered in the study, *e.g.*, hyperparameter-tuning to achieve energy efficiency of an AI algorithm;
- *Domain:* The domain considered in the study, *e.g.*, edge or mobile computing;
- *Type of data:* The type of data utilized by AI in the study, *e.g.*, text or images;
- *Artifact considered:* The AI artifact considered in the study, *e.g.*, the data used by AI models, the AI models themselves, or the AI deployment pipeline.
- *Considered phase:* If the study focused on the AI training phase, the AI inference phase, or both.
- *Research strategy:* The research strategy, as defined in [14], used to support the claims reported in the study;
- *Dataset size:* The size of the dataset, in number of data points, considered in the study (if any);
- *Energy Savings:* The reported percentage energy savings achieved by solutions reported in the study (if any is documented);
- *Industry involvement:* Industry involvement in the authorship of the study, which could be either academic-only authorship, industrial-only authorship, or mixed authorship;
- *Intended reader:* If the study is primarily intended for academic readers, industrial readers, or the general public.
- *Tool availability:* The availability of the tool(s) to address Green AI presented in the study (if any).

**2.2.5 Data Synthesis.** During the data extraction process, the data was harmonized by relying on the constant comparison [2] of extracted keywords, breaking up keywords into more specific ones

when their semantic depth required it, or merging very similar keywords to avoid redundancy. This analysis process relied on open coding [4] to systematically identify recurrent concepts, followed by axial coding [4] to manage the increasing complexity of some emerging concepts.

The only exceptions were made for the *research strategy*, *industry involvement*, and *tool availability* fields of the extraction framework (see Section 2.2.4), for which provisional coding was used [4]. Specifically, coding of the *research strategy* relied on the research strategy categories reported by Stol *et al.* [14] was used. The industry involvement instead relied on three pre-defined fields, namely “academic-only authorship”, “industrial-only authorship”, or “mixed authorship”. Finally, *tool availability* could only assume one of two pre-defined values, namely “Yes” (if the tool is available) or “No” (if the tool is not available, or none is presented in the primary study).

During the data extraction and synthesis phase, emerging codes are continuously discussed among the three authors of the review. This process ensures that the emerging codes and their abstraction level are kept consistent among researchers, and are aligned with the research goal and question of the study.

### 3 Results

In this section, we present the results collected with our SLR on Green AI.

#### 3.1 Publication Years

The literature spans from 2015 with the first publication on the topic to this present year (*i.e.*, 2022). Figure 2 presents the distribution of the literature papers regarding the publication year. We observe a global increase following the years. Furthermore, a spike in the number of publications is seen in 2020, going from 7 publications in 2019 to 20 in 2020. As the automated initial search was launched in 2022, the publication trends reported in this review might not be representative of the actual research output of 2022 (see also Section 2.2.1).

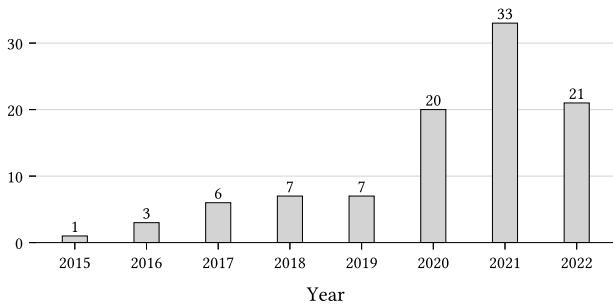


Figure 2: Number of publications per year.

#### 3.2 Venue Types

Publications are particularly concentrated on conferences ( $\triangleright 47$  out of 98 papers.) and journals ( $\triangleright 39$  out of 98 papers.). Only 12 out of the 98 publications are associated with a workshop. Conferences

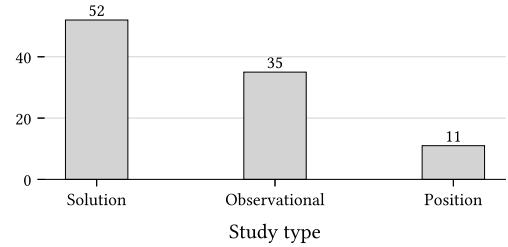


Figure 3: Number of publications per study type.

being treated as an equal publishing venue as journals follows the trends observed in the computer science research field [5, 16].

#### Green AI publication trends

• The topic of Green AI is experiencing an increasing trend of popularity, with a considerable growth in publications from 2020 onward. Most studies are published in conferences and journals, while only a minor portion in workshops.

#### 3.3 Study Types

Existing literature on Green AI spans across three types of studies, namely *observational*, *solution*, and *position* papers (see also Section 2.2.4). As shown in Figure 3, from the 98 papers covered in this review, the most common are solution papers, with 52 entries, followed by observational with 35, and position papers with 11. Note that study types are mutually exclusive, *i.e.*, a single paper has only one study type.

#### 3.4 Green AI Topics

From our analysis we identify 13 main topics being addressed by the Green AI literature. Figure 4 depicts the distribution of publications across the different topics. The most popular topic is *Monitoring*, addressed by 28 papers, followed by *Hyperparameter Tuning* (18), *Model Benchmarking* (17), *Deployment* (17), and *Model Comparison* (17). Since papers are not exclusive to a single topic, these top-4 topics alone cover 61% of the papers in this review. Below, we pinpoint each topic with a short summary and the respective number of publications.

**Monitoring**  $\triangleright$  28 out of 98 papers. Covering monitoring approaches to study the energy and/or carbon footprint of AI models.

In this topic, papers report and reflect on the energy footprint of state-of-the-art models throughout their lifecycle. For example, Wu *et al.* [106] provide a landscape of the carbon footprint of AI models across Facebook. Findings showcase that, typically, throughout the lifetime of AI models, 50% of their carbon cost lies in the embodied carbon footprint of the hardware used to develop these models. However, the paper shows that the vast majority of training workflows under-utilizes GPUs at 30–50% of their full capacity.

Other papers within this topic focus on solutions to make carbon monitoring feasible in any AI project [44]. As an example, the Carbontracker offers a toolset to track and predict the energy and carbon footprint of training DL models [22]. These studies argue

that it is quintessential to report the energy and carbon footprint of model development and training alongside performance metrics.

**Hyperparameter Tuning** ▷ **18 out of 98 papers**. Improving or assessing the impact on the energy consumption of optimizing hyperparameters when training an AI model.

Many publications are motivated by the fact that tuning parameters leads to significant energy costs – it requires retraining a model multiple times in order to find the optimal set of hyperparameter values. Hence, most publications within this topic focus on identifying alternative strategies that reduce the number of iterations required to tune hyperparameters [91].

On a different perspective, Chavannes *et al.* [83] explore how hyperparameter tuning can help deliver more energy-efficient models by adding power consumption to the set of parameters being optimized.

**Model Benchmarking** ▷ **17 out of 98 papers**. Studies that contribute with benchmarks to compare the energy footprint of different models or training techniques.

Benchmarks help the community understand how the state of the art behaves w.r.t. given performance indicators. Ultimately, they help create baselines so that new approaches can be properly validated and compared to the state of the art. As example of publications within this category, Asperti *et al.* [23] evaluate the energy cost of different variational autoencoders. Another study, by Yu *et al.* [113], compares the energy efficiency of common machine learning algorithms when applied to clinical laboratorial datasets.

**Deployment** ▷ **17 out of 98 papers**. Addressing the deployment stage of the lifecycle of an AI model.

Typically, publications in this topic discuss the problem of deploying AI models in a real scenario or in a scenario with peculiar constraints that challenge a standard approach. For example, deployment publications showcase the challenges of deploying energy-efficient AI in FPGA [97], in Edge devices [47, 64], in mobile devices [58, 75, 100], and so on.

**Precision/Energy Trade Off** ▷ **11 out of 98 papers**. There is a turning point where to increase a very small fraction of the model performance, it is required to endure an energy-intensive training loop. Within this topic, papers address the Pareto trade-off between having optimal accuracy and/or optimal energy efficiency.

Zhang *et al.* [114] study how removing neurons from neural networks affects both accuracy and energy consumption. Results indicate that a good portion of neurons are redundant and can be removed to reduce energy consumption without a significant impact on accuracy. At the same time, it shows that there is a turning point where removing neurons improves energy efficiency but significantly reduces accuracy. Hence, the two parameters always need to be analyzed together. Other works opt for optimizing energy while keeping accuracy loss within a negligible margin [102].

**Algorithm Design** ▷ **10 out of 98 papers**. Design of new training algorithms that produce models that are significantly more energy-efficient than the state of the art.

Some works propose small changes to the algorithms that make a big difference in the final energy consumption. For example, Garcia-Martin *et al.* [42] approximate the splitting criteria by selecting branches that require less computational effort. Results showcase

decision trees that are up to 31% more energy efficient and with minimal impact on accuracy. Other examples include Espnetv2 [77], a lightweight convolutional neural network designed with power-efficiency in mind.

**Libraries** ▷ **8 out of 98 papers**. Our choice of libraries have an impact on the final carbon footprint of AI systems. Studies within this topic provide some sort of evaluation of different AI libraries and how they contribute to energy efficiency.

This category shows that software engineering studies play an important role in enabling Green AI. Georgiou *et al.* [46] compare the energy footprint of deep learning frameworks. Results showcase that PyTorch is more energy-efficient than Tensorflow at the training stage. However, Tensorflow tends to be more energy-efficient at the inference stage. The study delves into the framework's different API methods and highlights code in the frameworks that should be optimized to reduce energy consumption. Finally, the authors motivate the importance of reporting and discussing energy efficiency in the documentation of deep learning frameworks.

**Data Centric** ▷ **6 out of 98 papers**. Typically, the AI community has looked into coming up with better model training strategies. However, there is a new trend in AI that is raising the importance of developing better data collection and processing techniques as a more effective way to deliver better AI models. This line of thought within Green AI aims at reducing the carbon footprint of AI by tackling the problem at the data level.

Data-centric approaches for Green AI show that feature selection and subsampling techniques can significantly reduce the energy consumption of training machine learning models [98]. Subsampling strategies can be more sophisticated by removing data points that are expected to be redundant in terms of knowledge acquisition [35].

**Network Architecture** ▷ **6 out of 98 papers**. The impact of a distributed network on the energy efficiency of AI. AI models are often deployed in a distributed context – e.g., IoT, edge computing, etc. Hence the design and architecture of the network plays an important role in leveraging sustainable models.

For example, Kim and Wu [64] propose an adaptive execution engine that selects the inference strategy according to the signal strength of the network in different devices, as it is known to affect the energy efficiency of the edge mobile system.

**Estimation** ▷ **5 out of 98 papers**. Collecting and making sense of energy or climate data is far from trivial – many different factors contribute to the final estimation [44]. This topic revolves around understanding ways of estimating the energy consumption or carbon footprint of models.

Existing solutions to estimate energy consumption for software fail to provide meaningful insight about energy consumption that can be mapped to a machine learning model's structure. IrEne creates a graph that breaks down NLP models into low-level machine learning primitives and provides energy estimations at the primitive level [33].

**Emissions** ▷ **4 out of 98 papers**. Papers that focus on understanding the carbon impact of creating and/or consuming AI systems. Dhar [36] flags the importance of being able to quantify carbon impact and the lack of tools and data available. Fraga-Lamas *et al.* [40]

go beyond reporting the energy consumption of an AI-enabled IoT scenario and present how much carbon would be emitted in different countries and different energy sources.

**Policy** ▷ **3 out of 98 papers.** Studies within this topic address and discuss strategies on how we should handle the carbon footprint of AI as a society.

Perucica and Andjelkovic [81] reflect on the environmental policies implemented by the European Union, discussing whether they fit the AI era or new regulations are needed. Rhode *et al.* [85] call out for the unclear dilemma between the impact of existing/upcoming AI technologies and the commitment to achieve the 1.5°C climate change goal as expressed in the UNFCCC Paris Declaration.

**Ethics** ▷ **3 out of 98 papers.** Papers that focus on the ethical implications of the growing carbon footprint of AI. Tamburini [96] discusses the responsibilities of AI scientists, AI infrastructure providers, and other stakeholders in enabling Green AI. The paper questions whether it is ethically justified to create massive AI pipelines to improve accuracy.

**Other** ▷ **5 out of 98 papers.** Studies addressing a relevant topic with only a single publication in total: User values [65], Scheduling [116], Rebound Effects [105], Security [89], Energy Capping [66].

#### Green AI topics by study type

⌚ There are 13 main topics on Green AI. The majority (61%) of the publications focuses on Monitoring, Hyperparameter-tuning, Model Benchmarking, and Deployment. Despite being important, topics such as Data-Centric, Estimation, and Emissions are underrepresented in the scientific literature.

### 3.5 Green AI Topics by Study Type

We further investigate the distribution of papers across different topics per category. Figure 5 presents a bubble plot that draws a bubble for each pair topic (x-axis) and study type (y-axis). The size of the bubble is proportional to the number of papers published in each pair. The plot enables a few observations.

Most topics adhere to the general pattern observed earlier in Section 3.3: the majority of papers consist of solution studies, followed by observational and then position. However, the topics of *Model Benchmarking* and *Libraries* do not follow this pattern, being mostly covered by observational papers. This is expected as these topics revolve around comparing different libraries and models to provide insight on the energy efficiency of different design decisions.

Moreover, papers from the least represented topics *Ethics*, *Policy*, and *Emissions* tend to be position papers. From the ten studies in these three topics, only one is observational and none is solution.

Also worth noticing is the fact that the majority of the position studies in Green AI only cover the smallest topics. Considering the top-10 topics – from Monitoring to Estimation – only 5 are position papers. In contrast, the bottom-4 topics (including Other) are covered by 10 position papers.

#### Green AI topics by study type

⌚ Most publications on Ethics, Policy, and Emissions are position studies calling for more research in these topics.

### 3.6 Domains

Figure 6 presents the distribution of the publications according to the domain they cover. The majority of the publications (*i.e.*, ▷ **58 out of 98 papers.**) do not devote their studies to a specific domain, but tackle the energy efficiency of AI in a general context. Regarding the most specific studies, the most covered domains are:

**Edge** Regarding Internet of Things and Edge Computing, which are usually associated with distributed systems and networks. ▷ **24 out of 98 papers.**

**Computer Vision** Regarding image recognition. ▷ **6 out of 98 papers.**

**Cloud** ▷ **5 out of 98 papers.**

**Mobile** ▷ **4 out of 98 papers.**

The **Other** category gathers publications about a specific domain, being covered only once, among Health, Autonomous Driving, Smart cities, Human Activity, Wearables, and Embedded Systems.

#### Green AI domains

⌚ The majority of Green AI studies does not focus a specific domain. Among specific domains, edge computing results to be the most recurrent one.

### 3.7 AI Pipeline Phases

The AI pipeline is divided into two major phases: the training, when the AI model is built, and the inference, when the model is used to make predictions from new data. Thus, we classify the papers according to 3 categories: **training**, **inference**, and **all**. The **all** category translates the fact that the paper does not consider a particular phase, but the whole pipeline.

As depicted in Figure 7, we find that most of the publications on the topics of Green AI focus on the training phase (▷ **49 out of 98 papers.**). In comparison, fewer papers direct their studies at the inference phase (▷ **17 out of 98 papers.**) or on the overall process (▷ **32 out of 98 papers.**).

#### Green AI Pipeline Phase

⌚ Approximately half of Green AI studies focus on the training phase, while a minor portion considers the entire AI pipeline. Only a minor portion of the Green AI literature focuses on the inference phase.

### 3.8 Considered Artifacts

AI systems are based on several artifacts, and tackling the energy efficiency of such systems can thus involve multiple of those artifacts (*e.g.*, data, model, pipeline) or different related artifacts (*e.g.*, architecture, framework, CPU). We study what particular artifacts are the focus of the trends in publication. We define categories according to those artifacts as:

**Model** The publications within this category focus on the model and/or associated algorithm to tackle the energy efficiency of AI. ▷ **63 out of 98 papers.**

**Data** Papers that address energy efficiency through the study of the data used in the AI pipeline. ▷ **8 out of 98 papers.**

**Pipeline** Studies looking at the whole AI pipeline. ▷ **3 out of 98 papers.**

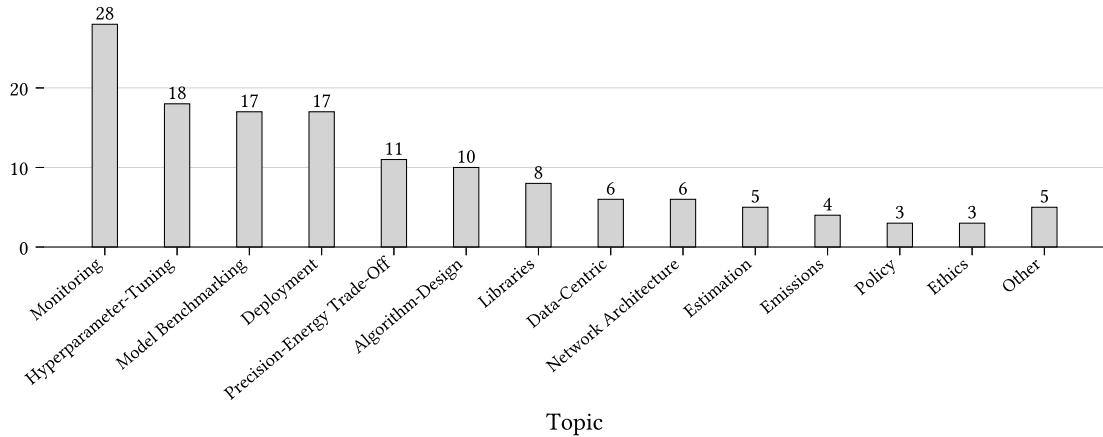


Figure 4: Number of papers per Green AI topic.

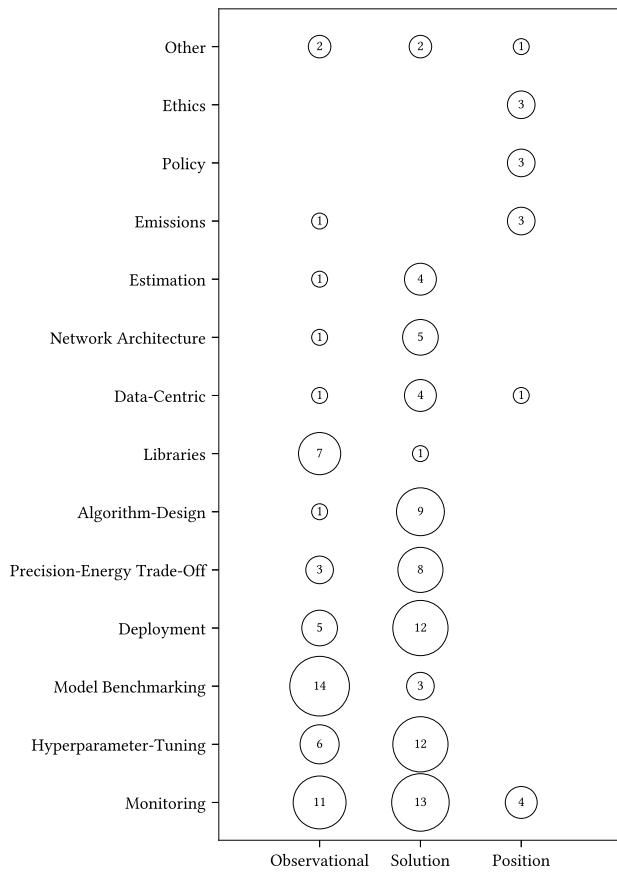


Figure 5: Number of publications by topic and study type.

**Other** Publications dealing with CPU, architecture, and framework.  $\triangleright 4$  out of 98 papers.

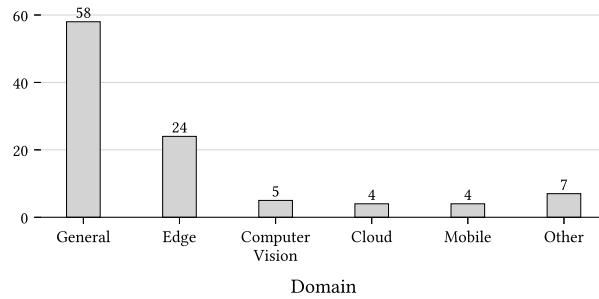


Figure 6: Number of publications per study domain.

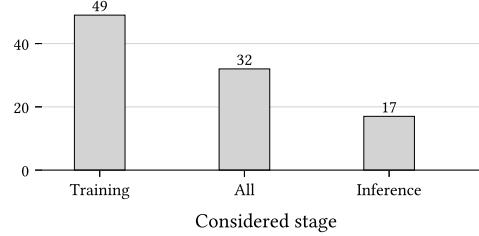
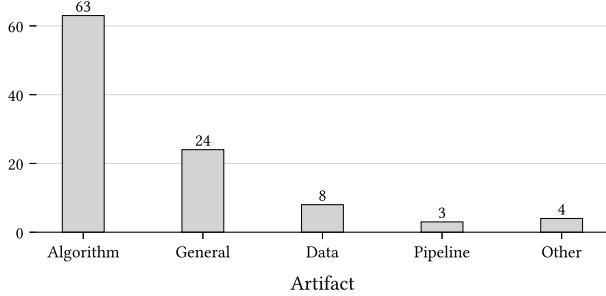


Figure 7: Number of publications per studied phase of AI.

**General** The papers do not specify a particular artifact and address AI systems as a whole.  $\triangleright 24$  out of 98 papers.

### 3.9 Algorithm Types

By considering the primary studies which focus on a specific algorithm ( $\triangleright 51$  out of 98 papers.), we note that the vast majority focus on **neural networks** ( $\triangleright 41$  out of 98 papers.). Only a much smaller fraction focuses on algorithms of different nature, such as decision trees ( $\triangleright 5$  out of 98 papers.), genetic algorithms ( $\triangleright 1$  out of 98 papers.), or logistic regression models ( $\triangleright 5$  out of 98 papers.).



**Figure 8: Number of publications per studied artifact.**

Regarding the deep neural network algorithms, we also note a further characterization of this field, with 8 studies focusing on **convolutional neural networks**, one on **transformers**, and one on **spiking neural networks**. We also observe three algorithms that appear only once in the Green AI literature (**Other** category,  $\triangleright 3$  out of 98 papers), namely *genetic algorithms*, *logic regression algorithms*, and *stochastic gradient descent algorithms*.

#### Green AI algorithm types

- Most Green AI primary studies are algorithm-agnostic or focus on neural networks. A small fraction uses decision trees.

### 3.10 Data Types Used

Regarding the types of data used in the Green AI body of literature, an overview of their distribution is reported in Figure 9. From the figure, we can observe that the recurrence of data types across primary studies is:

**Image data**  $\triangleright 42$  out of 98 papers.

**Textual data**  $\triangleright 22$  out of 98 papers.

**Numeric data**  $\triangleright 10$  out of 98 papers.

**Video data**  $\triangleright 4$  out of 98 papers.

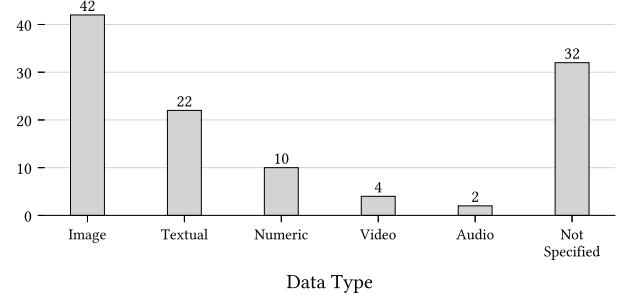
**Audio data**  $\triangleright 2$  out of 98 papers.

From the distribution of data types, we notice that image data is by far the most used one, and is utilized by almost half of the studies in the body of literature. The second most utilized data type is textual data, which nevertheless appears approximately half as often as the image one. Other types of data result to be less recurrent, with only few studies utilizing audio data (e.g., Lenherr et al. present a metric to measure the sustainability of Green AI by considering as case study the Intel MovidiusX processor, an embedded video processor with a Neural Engine for video processing and object detection [70]).

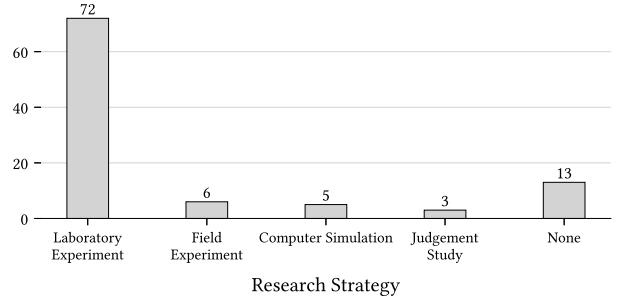
A rather high number of primary studies does not specify any kind of data (**Not specified** category,  $\triangleright 32$  out of 98 papers.). This finding has to be primarily attributed to the position and theoretical papers included in the review (see also Section 3.3 and Section 3.4).

#### Green AI data types

- Image data is the most used data type in Green AI studies, followed by textual and numeric data.



**Figure 9: Occurrence of data types used in the Green AI literature.**



**Figure 10: Occurrence of research strategies used in the Green AI literature.**

### 3.11 Dataset sizes

Regarding the size of the datasets used in the papers, approximately half of the primary studies ( $\triangleright 48$  out of 98 papers.) directly reference the number of data points used. By inspecting such numbers, we note that the number of data points used to study and to evaluate Green AI algorithms and approaches varies greatly, and ranges from **1k data points** [47] to **40M data points** [41]. Almost half of the studies reporting the number of data points ( $\triangleright 25$  out of 48 papers) utilize data points in the order of thousands ( $1k \leq \# datapoints \leq 70k$ ), while the remaining ( $\triangleright 23$  out of 48 papers) use one million data points or more ( $1M \leq \# datapoints \leq 40M$ ).

#### Green AI dataset sizes

- Dataset sizes range from 1k to 40M data points, with approximately half of the studies utilizing 1M or more data points.

### 3.12 Research Strategies

By considering the research strategies [14] utilized in the Green AI literature, the distribution of the various strategies, according to the collected primary studies, is reported in Figure 10.

The majority of paper results adopt **laboratory experiments** ( $\triangleright 72$  out of 98 papers.), while only a fraction uses other research strategies, such as **field experiments** ( $\triangleright 6$  out of 98 papers.), i.e.,

experiments conducted in pre-existing settings and **computer simulations**, *i.e.*, “*in silico*” simulations conducted in a nonempirical setting ( $\triangleright 5$  out of 98 papers.). As examples, Liu *et al.* [73] use a field study to assess a green software stack for computer vision of autonomous robots, while Yosuf *et al.* [112] leverage computer simulations to study how virtualized cloud fog networks can be used to improve AI energy efficiency.

#### Green AI Research Strategies

Most Green AI studies use laboratory experiments, while only a minority adopt other research strategies, such as field experiments and computer simulations.

### 3.13 Energy savings

By considering the energy savings reported achievable *via* Green AI strategies, we note that only approximately a third of the primary studies explicitly document them ( $\triangleright 27$  out of 98 papers.). Out of all Green AI strategies, among the ones which report concrete saving percentages, a technique based on structure simplification for deep neural networks results to save more energy, amounting to **115% energy savings** [114]. The other techniques which result to optimize energy the most are based on quantizing the inputs of decision trees [19] (97% energy savings), using data-centric Green AI techniques [98] (92% energy savings), and leveraging efficient deployment of AI algorithms *via* virtualized cloud fog networks (91% energy savings) [112]. Overall, more than half of the papers explicitly reporting energy saving percentages report a saving of at least 50% ( $\triangleright 17$  out of 27 papers), while only a minor number savings between 13% and 49%.

#### Green AI energy savings

Studies report energy savings between 13% and 115% energy savings, with more than half of the papers reporting savings of at least 50%.

### 3.14 Industry involvement

Regarding the industry involvement in Green AI scientific publications (see also Section 2.2.4), an overview of the authorship of the Green AI primary papers is depicted in Figure 11.

From the figure, we can note that most Green AI studies are authored exclusively by academic researchers ( $\triangleright 75$  out of 98 papers.), while also a considerable portion, amounting almost to a fourth of all primary studies, are authored by a mix of academic and industrial researchers ( $\triangleright 20$  out of 98 papers.). Green AI studies written exclusively by industrial authors appear only in rare instances ( $\triangleright 3$  out of 98 papers.).

#### Industry involvement

Most studies are written by academic authors, while a minor portion by a mix of academic and industrial authors. Green AI studies written exclusively by academic authors are very rare.

### 3.15 Intended readers

By considering the intended readers of the Green AI scientific literature, the vast majority targets academic readers ( $\triangleright 85$  out

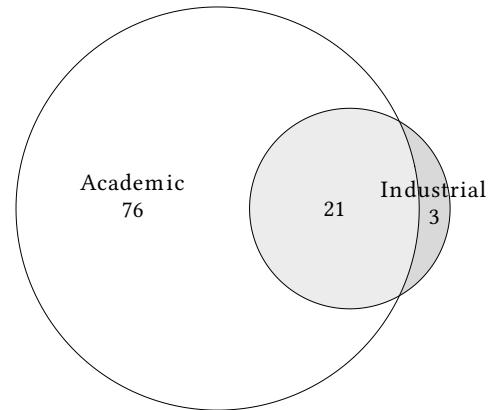


Figure 11: Industry involvement.

of 98 papers.), while a much smaller portion both academic and industrial readers ( $\triangleright 8$  out of 98 papers.). Despite scientific papers targetting intuitively a specialized audience, among the Green AI literature, few studies are intended also for the general public ( $\triangleright 5$  out of 98 papers.). For example, Dhar *et al.* [36], present an intuitive yet thoroughly positioned article on the systemic effect of AI on carbon emissions. Interestingly, among the primary studies, few are intended also for policymakers, *i.e.*, aim to sensibilize government stakeholders to consider issues related to Green AI. For example, in a paper by Rohde *et al.* [85], how opportunities and risks for the environment, economy and society associated with AI can be governed are discussed.

#### Intended readers

The vast majority of Green AI studies are targetting academic readers, while a much smaller portion targets both academic and industrial readers. A handful of studies, especially position papers, are intended for the general public.

### 3.16 Tool Provision

Among the primary studies collected for this literature review on Green AI, only a small fraction ( $\triangleright 15$  out of 98 papers.) makes tools available to tackle Green AI. The tools provided are of heterogeneous nature, and range from tools to monitor the resource efficiency of AI algorithms [48], to tools optimizing the energy efficiency for stochastic edge inference [64], and implementations of convolutional neural networks optimized for energy efficiency [77].

#### Green AI Tool Provision

Albeit numerous studies provide solution to tackle Green AI, only a fraction of them makes tools based on the solutions readily available online as an implemented tool.

## 4 Discussion

From the analysis of the Green AI publication trends a clear picture emerges. The topic is gaining increasing traction in the academic community, especially if the latest years are considered (from 2020

onward). Despite being a quite new research topic (with the first paper on Green AI being published in 2015), the socio-environmental relevance of the topic seems to be reflected in its targeted publication venues. With conferences and journal being the most recurrent Green AI publication venues, the Green AI research field seems to have positioned and consolidated itself quite quickly within AI research communities.

Regarding Green AI topics, the 13 different topics we discover in this review emphasize that Green AI is a broad field that needs to be tackled as a transdisciplinary field. Some topics are naturally tied to training strategies (e.g., monitoring, hyperparameter tuning, algorithm design). However, there are other topics that take Green AI outside the training realm.

This is the case for example of Deployment, Libraries, and Estimation that promise to be relevant in enabling Green AI. We argue that other disciplines need to be involved. For example, Software Engineering which has been dealing with these topics for traditional software systems. As highlighted by Cao *et al.* in their work on estimation[33], one cannot expect existing strategies for traditional software to address the new challenges of AI-based systems. Conversely, only a few Green AI papers [46, 52, 76] come from software engineering venues.

Our analysis also shows that the topics Estimation and Emissions are under-represented, with six and five papers, respectively. We argue that more work is quintessential in these topics to help scientists and practitioners report the carbon footprint of their AI models in a seamless way.

We showcase that papers under the topic Policy are only covered by position papers. We find this finding disconcerting: new policies to encourage Green AI within both industry and academic contexts need to be backed up with reliable evidence. Hence, we need more observational and solution papers that tackle this topic in the near future.

The same issue is present in Emissions – only one paper is observational and the remaining are position. It might be the case that computing the climate impact of AI is far from trivial and it is easier said than done. Again, this is a call for the community to take action. It is not enough to ask big companies to provide their data on carbon impact – we also need to provide strategies and solutions to make it standard and straightforward.

From the results collected regarding the Green AI domains we deduce that, in order to improve the environmental sustainability of AI, it is often not necessary to focus on a specific domain. This implies that frequently fundamental aspects of Green AI are still open to investigation, and results can then be ported from a generic setting to specific domains. However, from the obtained results, we also note that the increasing distribution of digital infrastructures to achieve environmental sustainability [15] might have played a role in Green AI research, with edge computing being the most considered specific domain.

The results regarding the AI pipeline phases considered in the literature unequivocally point to training as the most studied phase. Albeit the training phase is intuitively the most energy-greedy phase, this results calls for a word of caution. From recent results (e.g., a study on data-centric Green AI [98]) the inference phase results to consume only a negligible fraction of the energy consumed in the training phase. Nevertheless, given the high execution rate

of the inference phase, how the energy consumed by the infrequent execution of the training phase compares to one of the highly executed inference phase is still an open question. As a call for action, studies should be conducted by considering the energy consumed throughout the whole life cycle of AI models, from their training to inference phase, till their eventual depreciation.

By considering the data types used in Green AI studies, we note that the vast majority of the literature uses image data. To the best of our knowledge, this choice is not guided by any specific research design choice (e.g., AI models based on image data being the most used in practice, or being the most energy greedy ones). For this reason, we conjecture that the popularity of utilizing image data for Green AI data is mostly driven by convenience, either because past work focused on such data by chance, image datasets are more accessible/standardized with respect to other ones, or more off the shelf image AI models/libraries are currently available. Regardless of the cause, this result points to the need of utilizing more heterogeneous data types, rather than focusing primarily on image data, in order to gain a holistic understanding of Green AI.

The most common research strategy adopted for Green AI studies clearly emerges from the literature as being laboratory experiments. Given the fast popularization and consolidation of the Green AI research field, from this review it seems as if the time is suitable to shift the focus to other research strategies, e.g., field experiments and case studies. This would not only allow to change the considered context from an *in vitro* to an *in vivo* setting, but also to bridge potential gaps between academic research and industrial practice.

From the results of this review regarding energy savings, we deduce that the research field of Green AI is highly promising, with more than half of the papers reporting 50% or more energy savings. This study focuses on the state of the art of Green AI, rather than focusing on the state of practice. It would be therefore interesting to understand, as future work, the extent to which this encouraging results are transposed to industrial practice, and the potential impediments which hinder their adoption or full potential.

Regarding industry involvement in Green AI studies, the results gathered from this review are promising. The authorship of Green AI literature results to be to a good extent shared between academic and industrial researchers/practitioners. This finding might highlight the sensibility of industry towards Green AI concerns, and/or the importance of moving towards more environmentally sustainable AI practices.

As a potential impediment to the industrial adoption of Green AI research, our results point to a low recurrence of studies targeted towards practitioners. While numerous journals are explicitly aimed at practitioners, e.g., IEEE Software<sup>2</sup>, only few studies on Green AI included in our review target them. This result might point to the fact that the Green AI interest is still primarily focused towards academic activities, while the authorship showcases a rather high interest of industry. As take away, similar to the considerations made for the Green AI research strategies, it might be the right moment to consider a higher involvement of industry in Green AI, which results to date to be a research area still targeted primarily towards academic readers.

---

<sup>2</sup><https://www.computer.org/csdl/magazine/so>. Accessed 22nd December 2022.

Finally, from this review, we note that the current situation regarding the provisioning of Green AI tools is not bright. Albeit the majority of the studies present Green AI solutions, only a small fraction of them makes the solutions available as a tool. We conjecture that this result might either point towards (i) a fast-paced nature of Green AI research, in which results are rapidly deprecated, and hence tools are not meaningful, or (ii) an immaturity of the research field, which still requires a solid empirical foundation on which tools can be built upon.

## 5 Threats to Validity

In this section, we discuss the threats to validity of our study. To ensure the quality of the results, we established a well-defined research protocol to proceed with the data collection. In addition, throughout our study, we followed the recommendations of the guidelines for conducting a systematic literature review [6–8, 12, 17]. We designed and carried the different reviewing processes according to the rigorous protocol we established after the guidelines and described in Section 2. Nevertheless, some threats to validity can still exist even with our best efforts. In the following, we present the threats which could have influenced our study, jointly with the strategies we adopted to mitigate them.

**External validity** The main threat to external validity is that the literature collected and analysed in this study is not sufficiently representative. To avoid this situation, we surveyed three prominent literature indexers through an automatic query (*i.e.*, Google Scholar, Scopus, and Web of Science), and left the year of publication unbounded, to reduce the probability of missing any relevant publication. In addition, the search query was designed to target relevant literature directly with specific keywords, while allow for flexibility by considering similar, complementary, and variation of the keywords (*e.g.*, the keywords *green*, *sustainability*, and *sustainable*). We also mitigated the threat of having an incomplete set of studies, as well as the threat associated with the specificity of the terms used in the search query, by performing a complementary iterative bidirectional snowballing process of the query results. This latter search strategy allowed us to include literature related to our query that was not directly referencing any of the automated search keywords. We limited our review of the literature to peer-reviewed studies, to moderate the threat about the low quality of the set of primary studies. We deem that such practice does not constitute an additional threat, as peer-review is a standard requirement of high-quality publications.

**Internal validity** To address potential threats to internal validity, we established a rigorous research protocol *a priori*, and we followed it to conduct all the research activities. Subjective biases and interpretations were mitigated by closely complying with the selection criteria to evaluate the studies. Moreover, weekly meeting were held during the selection process to jointly discuss examples, doubts, and to align the selection process between the three researchers.

**Construct validity** To ensure that the set of studies answered our research questions, we applied *a priori* carefully constructed

inclusion and exclusion criteria to strictly control the manual selection of studies. We then used the bidirectional snowballing technique to expand the range of relevant primary studies to a more comprehensive set.

**Conclusion validity** Possible sources of bias arising from the data extraction and analysis phases were mitigated by strict compliance with an *a priori* defined protocol, explicitly tailored to collect the data needed to answer our research questions. In all, we followed the best practises of the standard guidelines for systematic literature reviews [6–8, 12, 17]. Lastly, we documented all the data throughout the whole review process and made them available for reproducibility and replicability purposes (see Section 1).

## 6 Related Work

Despite the growing interest around Green AI, the topic has been marginally considered only in a handful of reviews. The related work mainly investigates the topic as an intersection of AI and environmental sustainability, or by defining it as a specific subdomain of software engineering. To the best of our knowledge, this review is the first aiming towards a comprehensive review of Green AI research and its characteristics.

In a recent publication, Natarajan *et al.* perform a systematic literature review on the topics of ‘AI for Environmental Sustainability’ as well as ‘Environmental Sustainability of AI’. The authors present the affordances of the use of AI for sustainability that they extracted from the literature [10]. ‘AI affordances’ are introduced as the possible actions offered by AI artifacts to an organizational actor whose goal is to achieve environmental sustainability. The authors point out the focus of previous research on the technical side, and they advocate for a further exploration of the concept of sustainable AI affordances from a socio-technical perspective. The literature is exclusively analyzed with respect to building the AI affordances, and other characteristics of the state-of-the art of Green AI are considered nor discussed in the study. In contrast, our review focuses on the sustainability of AI, and maps the entirety of the Green AI literature. In our review, we aim at providing a detailed and comprehensive overview of the characteristics of the Green AI state-of-the-art research (*e.g.*, topic, domain, type of study, targeted artifact, overview of energy savings, tool provision, industrial involvement). Therefore, in contrast to the work of Natarajan *et al.* [10], we consider the different facets of Green AI, rather than exclusively on AI affordances, leading to a more holistic review of Green AI, and a higher number of considered primary studies (98 versus 41 papers). This difference could be explained by the fact that their review only includes papers involving consumer products and services and excludes papers dealing with non-commercial applications, whereas we provide an overview of the whole field of Green AI.

Previous literature reviews consider Green AI research by focusing exclusively on specific subdomains of AI and application subdomains of Software Engineering, *e.g.*, deep learning [18], information retrieval [13], or embedded systems [9]. In contrast, our research aims to review the entirety of the Green AI literature, regardless of the specific AI or software engineering subdomain it focuses on.

In the survey of Xu et al. [18], the authors provide an overview of the approaches aimed at improving the environmental sustainability of deep learning. The authors map the different approaches using a taxonomy of the deep learning life cycle stage and its related artifacts. In contrast to such study, in this review we target a higher number of Green AI characteristics (see Section 2.2.4), and target the entirety of Green AI literature, rather than exclusively the one on deep learning.

Scells et al. [13] provide a literature review on methods related to the domain of Green Information Retrieval. The authors explain that the domain of Information Retrieval (IR) produces relatively low emissions compared to other research domains, but they also warn that similar trends of costs and environmental impact may appear considering the growing development of new IR-focused deep learning models. Natural Language Processing and Machine Learning are also discussed, but only with respect to the Information Retrieval domain. Therefore, they are not addressing the whole field of AI, as done in this review.

Finally, the optimizations that can be made for the implementation of deep learning models on the specific platform of NVIDIA Jetson are reviewed with a focus on energy efficiency by Mittal [9]. The review covers studies at both the hardware and software level. Nevertheless, the review addresses only the Jetson platform<sup>3</sup>. We differentiate ourselves from this study by providing a holistic review of Green AI, rather than focusing exclusively on deep learning.

## 7 Conclusion

In this systematic literature review, we aimed at characterizing the existing body of research in Green AI. We identified 98 peer-reviewed publications that show a significant growth in this research field since 2020.

We provide an encompassing overview and characterization of the different topics being addressed by Green AI papers. We identified 13 different Green AI topics, showcasing that the spotlight falls on monitoring, hyperparameter-tuning, model benchmarking, and deployment. Less frequent topics – such as data-centric, estimation, and emissions – show less obvious approaches that deserve further research in the upcoming years.

The potential of Green AI cannot be disregarded: the majority of publications show significant energy savings, up to 115%, at little or no cost in accuracy. However, we argue that most publications revolve around laboratory studies. More field experiments are quintessential to help AI practitioners embrace green strategies that are effective, feasible, and measurable. This is also reflected in the small participation of the industry in these studies – only 23% of publications involve industry partners.

At the same time, we conclude that the field seems to be reaching a considerable level of maturity. Hence, it is necessary to encourage the port of promising academic results to industrial practice. In other words, our study calls out for the importance of having reproducible research. Only a small fraction of solution papers offers a tool or software package that can be used by the community. We argue that Green AI is an urgent and necessary line of research that needs to grow fast and solid – non-replicable research can only slow us down.

<sup>3</sup><https://developer.nvidia.com/embedded-computing>. Accessed 23th December 2022.

This review also serves as a foundation for future research that ultimately aims to reduce the climate impact of AI. In this respect, we see potential in follow-up grey literature or interview studies to understand how AI professionals are currently addressing the issue.

## References

- [1] Victor R. Basili, Gianluigi Caldiera, and Dieter Rombach. 1994. The Goal Question Metric Approach. In *Encyclopedia of Software Engineering*. Wiley, 528–532.
- [2] Barney G Glaser. 1965. The constant comparative method of qualitative analysis. *Social problems* 12, 4 (1965), 436–445.
- [3] Mark Haakman, Luís Cruz, Hennie Huijgens, and Arie van Deursen. 2021. AI lifecycle models need to be revised. *Empirical Software Engineering* 26, 5 (2021), 1–29.
- [4] Bryan Jenner, Uwe Flick, Ernst von Kardoff, and Ines Steinke. 2004. *A companion to qualitative research*. Sage, 271–275.
- [5] Jinseok Kim. 2019. Author-based analysis of conference versus journal publication in computer science. *J. Assoc. Inf. Sci. Technol.* 70, 1 (Jan. 2019), 71–82. <https://doi.org/10.1002/asi.24079>
- [6] Barbara Kitchenham. 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University* 33, TR/SE-0401 (2004), 28.
- [7] Barbara Kitchenham and Pearl Brereton. 2013. A systematic review of systematic review process research in software engineering. *Information and software technology* 55, 12 (2013), 2049–2075.
- [8] Philipp Mayring et al. 2004. Qualitative content analysis. *A companion to qualitative research* 1, 2 (2004), 159–176.
- [9] Sparsh Mittal. 2019. A Survey on optimized implementation of deep learning models on the NVIDIA Jetson platform. *J. Syst. Archit.* 97 (Aug. 2019), 428–442. <https://doi.org/10.1016/j.sysarc.2019.01.011>
- [10] Harish Karthi Natarajan, Danielly de Paula, Christian Dremel, and Falk Uebenickel. 2022. A Theoretical Review on AI Affordances for Sustainability. In *Americas Conference on Information Systems*.
- [11] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. 2008. Systematic mapping studies in software engineering. *International Conference on Evaluation and Assessment in Software Engineering*, 68–77.
- [12] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and software technology* 64 (2015), 1–18.
- [13] Harrisson Scells, Shengyao Zhuang, and Guido Zuccon. 2022. Reduce, Reuse, Recycle: Green Information Retrieval Research. In *SIGIR '22: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, New York, NY, USA, 2825–2837. <https://doi.org/10.1145/3477495.3531766>
- [14] Klaas-Jan Stol and Brian Fitzgerald. 2018. The ABC of software engineering research. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 27, 3 (2018), 1–51.
- [15] Roberto Verdecchia, Patricia Lago, and Carol de Vries. 2022. The future of sustainable digital infrastructures: A landscape of solutions, adoption factors, impediments, open problems, and scenarios. *Sustainable Computing: Informatics and Systems* (2022), 100767.
- [16] George Vrettas and Mark Sanderson. 2015. Conferences versus journals in computer science. *J. Assoc. Inf. Sci. Technol.* 66, 12 (Dec. 2015), 2674–2684. <https://doi.org/10.1002/asi.23349>
- [17] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *International Conference on Evaluation and Assessment in Software Engineering*. ACM Press, 1–10.
- [18] Jingjing Xu, Wangchunshu Zhou, Zhiyi Fu, Hao Zhou, and Lei Li. 2021. A Survey on Green Deep Learning. *arXiv* (Nov. 2021). <https://doi.org/10.48550/arXiv.2111.05193>

## Primary Studies

- [19] Bruno Abreu, Mateus Grellert, and Sergio Bampi. 2020. VLSI design of tree-based inference for low-power learning applications. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–5.
- [20] Bruno Abreu, Mateus Grellert, and Sergio Bampi. 2022. A framework for designing power-efficient inference accelerators in tree-based learning applications. *Engineering Applications of Artificial Intelligence* 109 (2022), 104638.
- [21] Phyllis Ang, Bhuwan Dhangra, and Lisa Wu Wills. 2022. Characterizing the Efficiency vs. Accuracy Trade-off for Long-Context NLP Models. In *Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP*. Association for Computational Linguistics, Dublin, Ireland, 113–121. <https://doi.org/10.18653/v1/2022.nlppower-1.12>
- [22] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. 2020. Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep

- Learning Models. ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems. arXiv:2007.03051.
- [23] Andrea Aspert, Davide Evangelista, and Elena Loli Piccolomini. 2021. A Survey on Variational Autoencoders from a Green AI Perspective. *SN Comput. Sci.* 2, 4 (July 2021), 1–23. <https://doi.org/10.1007/s42979-021-00702-9>
- [24] Nesrine Bannour, Sahar Ghannay, Aurélie Névéol, and Anne-Laure Ligozat. 2021. Evaluating the carbon footprint of NLP methods: a survey and analysis of existing tools. In *EMNLP, Workshop SustaiNLP*.
- [25] Michel Barlaud and Frédéric Guyard. 2021. Learning sparse deep neural networks using efficient structured projections on convex constraints for green AI. In *2020 25th International Conference on Pattern Recognition (ICPR)*. 1566–1573. <https://doi.org/10.1109/ICPR48806.2021.9412162>
- [26] Soroush Bateni, Husheng Zhou, Yuankun Zhu, and Cong Liu. 2018. Predjoule: A timing-predictable energy optimization framework for deep neural networks. In *2018 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 107–118.
- [27] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, New York, NY, USA, 610–623. <https://doi.org/10.1145/3442188.3445922>
- [28] Alexander El Brownlee, Jason Adair, Saemundur O Haraldsson, and John Jabbio. 2021. Exploring the accuracy–energy trade-off in machine learning. In *2021 IEEE/ACM International Workshop on Genetic Improvement (GI)*. IEEE, 11–18.
- [29] Sevda Ozge Bursa, Ozlem Durmaz Incel, and Gulfem Isiklar Alptekin. 2022. Transforming Deep Learning Models for Resource-Efficient Activity Recognition on Mobile Devices. In *2022 5th Conference on Cloud and Internet of Things (CIoT)*. IEEE, 83–89.
- [30] Ermao Cai, Da-Cheng Juan, Dimitrios Stamoulis, and Diana Marculescu. 2017. *NeuralPower: Predict and Deploy Energy-Efficient Convolutional Neural Networks*. In *Proceedings of the Ninth Asian Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 77)*. Min-Ling Zhang and Yung-Kyun Noh (Eds.). PMLR, Yonsei University, Seoul, Republic of Korea, 622–637. <https://proceedings.mlr.press/v77/cai17a.html>
- [31] Antonio Candieri, Riccardo Perego, and Francesco Archetti. 2021. Green machine learning via augmented Gaussian processes and multi-information source optimization. *Soft Comput.* 25, 19 (Oct. 2021), 12591–12603. <https://doi.org/10.1007/s00500-021-05684-7>
- [32] Qingqing Cao, Aruna Balasubramanian, and Niranjan Balasubramanian. 2020. Towards Accurate and Reliable Energy Measurement of NLP Models. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*. Association for Computational Linguistics, Online, 141–148. <https://doi.org/10.18653/v1/2020.sustainlp-1.19>
- [33] Qingqing Cao, Yash Kumar Lal, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. 2021. IrEne: Interpretable Energy Prediction for Transformers. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (2021). <https://doi.org/10.18653/v1/2021.acl-long.167>
- [34] Francisco M Castro, Nicolás Gui, Manuel J Marín-Jiménez, Jesús Pérez-Serrano, and Manuel Ujaldón. 2019. Energy-based tuning of convolutional neural networks on multi-GPUs. *Concurrency and Computation: Practice and Experience* 31, 21 (2019), e4786.
- [35] Priyadarshan Dhabe, Param Mirani, Rahul Chugwani, and Sadanand Ganderwar. 2021. Data Set Reduction to Improve Computing Efficiency and Energy Consumption in Healthcare Domain. In *Digital Literacy and Socio-Cultural Acceptance of ICT in Developing Countries*. Springer, 53–64.
- [36] Payal Dhar. 2020. The carbon impact of artificial intelligence. *Nat. Mach. Intell.* 2, 8 (2020), 423–425.
- [37] Josefa Diaz-Álvarez, Pedro A Castillo, Francisco Fernández de Vega, Francisco Chávez, and Jorge Alvarado. 2022. Population size influence on the energy consumption of genetic programming. *Measurement and Control* (2022), 00202940211064471.
- [38] Jesse Dodge, Taylor Prewitt, Remi Tachet des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Lucioni, Noah A. Smith, Nicole DeCarlo, and Will Buchanan. 2022. Measuring the Carbon Intensity of AI in Cloud Instances. In *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, New York, NY, USA, 1877–1894. <https://doi.org/10.1145/3531146.3533234>
- [39] Mariza Ferro, Gabrieli D. Silva, Felipe B. de Paula, Vitor Vieira, and Bruno Schulze. 2021. Towards a sustainable artificial intelligence: A case study of energy efficiency in decision tree algorithms. *Concurrency Computat. Pract. Exper.* n/a, n/a (Dec. 2021), e6815. <https://doi.org/10.1002/cpe.6815>
- [40] Paula Fraga-Lamas, Sérgio Ivan Lopes, and Tiago M. Fernández-Caramés. 2021. Green IoT and Edge AI as Key Technological Enablers for a Sustainable Digital Transition towards a Smart Circular Economy: An Industry 5.0 Use Case. *Sensors* 21, 17 (Aug. 2021), 5745. <https://doi.org/10.3390/s21175745>
- [41] Eva García-Martin, Niklas Lavesson, and Håkan Grahn. 2017. Identification of Energy Hotspots: A Case Study of the Very Fast Decision Tree. In *Green, Pervasive, and Cloud Computing*. Springer, Cham, Switzerland, 267–281. [https://doi.org/10.1007/978-3-319-57186-7\\_21](https://doi.org/10.1007/978-3-319-57186-7_21)
- [42] Eva García-Martin, Niklas Lavesson, Håkan Grahn, Emiliano Casalicchio, and Veselka Boeva. 2021. Energy-aware very fast decision tree. *Int. J. Data Sci. Anal.* 11, 2 (March 2021), 105–126. <https://doi.org/10.1007/s41060-021-00246-4>
- [43] Ángel M. García-Vico and Francisco Herrera. 2021. A Preliminary Analysis on Software Frameworks for the Development of Spiking Neural Networks. In *Hybrid Artificial Intelligent Systems*. Springer, Cham, Switzerland, 564–575. [https://doi.org/10.1007/978-3-030-86271-8\\_47](https://doi.org/10.1007/978-3-030-86271-8_47)
- [44] Eva García-Martin, Crefeda Faviola Rodrigues, Graham Riley, and Håkan Grahn. 2019. Estimation of energy consumption in machine learning. *J. Parallel and Distrib. Comput.* 134 (2019), 75–88.
- [45] Koen Gauen, Rohit Rangan, Anup Mohan, Yung-Hsiang Lu, Wei Liu, and Alexander C. Berg. 2017. Low-power image recognition challenge. In *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. 99–104. <https://doi.org/10.1109/ASPDAC.2017.7858303>
- [46] Stefanos Georgiou, Maria Kechagia, Tushar Sharma, Federica Sarro, and Ying Zou. 2022. Green AI: do deep learning frameworks have different costs? In *ICSE '22: Proceedings of the 44th International Conference on Software Engineering*. Association for Computing Machinery, New York, NY, USA, 1082–1094. <https://doi.org/10.1145/3510003.3510221>
- [47] Santosh Gondi and Vineel Pratap. 2021. Performance and Efficiency Evaluation of ASR Inference on the Edge. *Sustainability* 13, 22 (Nov. 2021), 12392. <https://doi.org/10.3390/su132212392>
- [48] Achim Guldner, Sandro Kreten, and Stefan Naumann. 2021. Exploration and systematic assessment of the resource efficiency of Machine Learning.. In *GI-Jahrestagung*. 287–299.
- [49] Abhishek Gupta, Camille Lanteigne, and Sara Kingsley. 2020. SECure: A Social and Environmental Certificate for AI Systems. *ICML 2020 Challenges in Deploying and monitoring Machine Learning Systems Workshop* (June 2020). <https://doi.org/10.48550/arXiv.2006.06217>
- [50] María Gutiérrez, Ma Ángeles Moraga, and Félix García. 2022. Analysing the energy impact of different optimisations for machine learning models. In *2022 International Conference on ICT for Sustainability (ICT4S)*. IEEE, 46–52.
- [51] Başak Güler and Aylin Yener. 2021. Energy-Harvesting Distributed Machine Learning. In *2021 IEEE International Symposium on Information Theory (ISIT)*. 320–325. <https://doi.org/10.1109/ISIT45174.2021.9518045>
- [52] Raluca Maria Hampau, Maurits Kaptein, Robin van Emden, Thomas Rost, and Ivano Malavolta. 2022. An Empirical Study on the Performance and Energy Consumption of AI Containerization Strategies for Computer-Vision Tasks on the Edge. In *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering 2022* (Gothenburg, Sweden) (EASE '22). Association for Computing Machinery, New York, NY, USA, 50–59. <https://doi.org/10.1145/3530019.3530025>
- [53] Walid A. Hanafy, Tergel Molom-Ochir, and Rohan Shenoy. 2021. Design Considerations for Energy-efficient Inference on Edge Devices. In *e-Energy '21: Proceedings of the Twelfth ACM International Conference on Future Energy Systems*. Association for Computing Machinery, New York, NY, USA, 302–308. <https://doi.org/10.1145/3447555.3465326>
- [54] Soheil Hashemi, Nicholas Anthony, Hokchhay Tann, R Iris Bahar, and Sherief Reda. 2017. Understanding the impact of precision quantization on the accuracy and energy of neural networks. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017. IEEE, 1474–1479.
- [55] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning. *Journal of Machine Learning Research* 21, 248 (2020), 1–43. <http://jmlr.org/papers/v21/20-312.html>
- [56] Miro Hodak and Ajay Dholakia. 2021. Recent Efficiency Gains in Deep Learning: Performance, Power, and Sustainability. In *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2040–2045.
- [57] Hamzaoui Ikhlasse, Duthil Benjamin, Courboulay Vincent, and Medromi Hicham. 2022. Recent implications towards sustainable and energy efficient AI and big data implementations in cloud-fog systems: A newsworthy inquiry. *Journal of King Saud University - Computer and Information Sciences* 34, 10, Part A (Nov. 2022), 8867–8887. <https://doi.org/10.1016/j.jksuci.2021.11.002>
- [58] Nitthilan Kanappan Jayakodi, Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. 2020. Design and optimization of energy-accuracy tradeoff networks for mobile platforms via pretrained deep models. *ACM Transactions on Embedded Computing Systems (TECS)* 19, 1 (2020), 1–24.
- [59] Wandri Jooste, Rejwanul Haque, and Andy Way. 2022. Knowledge Distillation: A Method for Making Neural Machine Translation More Efficient. *Information* 13, 2 (Feb. 2022), 88. <https://doi.org/10.3390/info13020088>
- [60] Sorin Liviu Jurj, Flavius Opritoiu, and Mircea Vladutiu. 2020. Environmentally-friendly metrics for evaluating the performance of deep learning models and systems. In *International Conference on Neural Information Processing*. Springer, 232–244.
- [61] Petra Jääskeläinen, Daniel Pargman, and André Holzapfel. 2022. On the environmental sustainability of AI art(s). In *Eighth Workshop on Computing within*

- Limits.* <https://doi.org/10.21428/bf6fb269.c46375fa>
- [62] Lynn H. Kaack, Priya L. Dotti, Emma Strubell, George Kamiya, Felix Creutzig, and David Rolnick. 2022. Aligning artificial intelligence with climate change mitigation. *Nat. Clim. Change* 12 (June 2022), 518–527. <https://doi.org/10.1038/s41558-022-01377-7>
- [63] Minsu Kim, Walid Saad, Mohammad Mozaffari, and Merouane Debbah. 2022. On the Tradeoff between Energy, Precision, and Accuracy in Federated Quantized Neural Networks. In *ICC 2022 - IEEE International Conference on Communications*. 2194–2199. <https://doi.org/10.1109/ICCC45855.2022.9838362>
- [64] Young Geun Kim and Carole-Jean Wu. 2020. Autoscale: Energy efficiency optimization for stochastic edge inference using reinforcement learning. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 1082–1096.
- [65] Pascal D. König, Stefan Wurster, and Markus B. Siewert. 2022. Consumers are willing to pay a price for explainable, but not for green AI. Evidence from a choice-based conjoint analysis. *Big Data & Society* 9, 1 (Jan. 2022), 20539517211069632. <https://doi.org/10.1177/20539517211069632>
- [66] Adam Krzywaniak, Paweł Czarnul, and Jerzy Proficz. 2022. GPU Power Capping for Energy-Performance Trade-Offs in Training of Deep Convolutional Neural Networks for Image Recognition. In *International Conference on Computational Science*. Springer, 667–681.
- [67] Mohit Kumar, Xingzhou Zhang, Liangkai Liu, Yifan Wang, and Weisong Shi. 2020. Energy-Efficient Machine Learning on the Edges. In *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. 912–921. <https://doi.org/10.1109/IPDPSW50202.2020.00153>
- [68] Jaeha Kung, Duckhwan Kim, and Saibal Mukhopadhyay. 2015. A power-aware digital feedforward neural network platform with backpropagation driven approximate synapses. In *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 85–90.
- [69] Loïc Lannelongue, Jason Grealey, and Michael Inouye. 2021. Green Algorithms: Quantifying the Carbon Footprint of Computation. *Adv. Sci.* 8, 12 (June 2021), 2100707. <https://doi.org/10.1002/advs.202100707>
- [70] Nicola Lenherr, René Pawlitzek, and Bruno Michel. 2021. New universal sustainability metrics to assess edge intelligence. *Sustainable Computing: Informatics and Systems* 31 (Sept. 2021), 100580. <https://doi.org/10.1016/j.suscom.2021.100580>
- [71] Da Li, Xinbo Chen, Michela Bechini, and Ziliang Zong. 2016. Evaluating the Energy Efficiency of Deep Convolutional Neural Networks on CPUs and GPUs. In *2016 IEEE International Conferences on Big Data and Cloud Computing (BD-Cloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom)*. 477–484. <https://doi.org/10.1109/BDCloud-SocialCom-SustainCom.2016.76>
- [72] Anne-Laure Ligozat, Julien Lefevre, Aurélie Bugeau, and Jacques Combaz. 2022. Unraveling the Hidden Environmental Impacts of AI Solutions for Environment Life Cycle Assessment of AI Solutions. *Sustainability* 14, 9 (April 2022), 5172. <https://doi.org/10.3390/su14095172>
- [73] Liangkai Liu, Jiamin Chen, Marco Brocanelli, and Weisong Shi. 2019. E2M: an energy-efficient middleware for computer vision applications on autonomous mobile robots. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. 59–73.
- [74] Michele Magno, Michael Pritz, Philipp Mayer, and Luca Benini. 2017. Deep-Emote: Towards multi-layer neural networks in a low power wearable multi-sensors bracelet. In *2017 7th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI)*. IEEE, 32–37.
- [75] Susmita Dey Manasi, Farhana Sharmin Snigdha, and Sachin S Sapatnekar. 2020. NeuPart: Using analytical models to drive energy-efficient partitioning of CNN computations on cloud-connected mobile clients. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28, 8 (2020), 1844–1857.
- [76] Andrea McIntosh, Safwat Hassan, and Abram Hindle. 2019. What can Android mobile app developers do about the energy consumption of machine learning? *Empir. Software Eng.* 24, 2 (April 2019), 562–601. <https://doi.org/10.1007/s10664-018-9629-2>
- [77] Sachin Mehta, Mohammad Rastegari, Linda Shapiro, and Hannaneh Hajishirzi. 2019. Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9190–9200.
- [78] Thaha Mohammed, Aiiad Albeshri, Iyad Katib, and Rashid Mehmood. 2020. UbiPriSEQ—Deep Reinforcement Learning to Manage Privacy, Security, Energy, and QoS in 5G IoT HetNets. *Appl. Sci.* 10, 20 (Oct. 2020), 7120. <https://doi.org/10.3390/app10207120>
- [79] Elena Morotti, Davide Evangelista, and Elena Loli Piccolomini. 2021. A green prospective for learned post-processing in sparse-view tomographic reconstruction. *Journal of Imaging* 7, 8 (2021), 139.
- [80] David Patterson, Joseph Gonzalez, Urs Hözle, Quoc Le, Chen Liang, Lluís-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. 2022. The Carbon Footprint of Machine Learning Training Will Plateau. Then Shrink. *Computer* 55, 7 (2022), 18–28. <https://doi.org/10.1109/MC.2022.3148714>
- [81] Natasja Perucica and Katarina Andjelkovic. 2022. Is the future of AI sustainable? A case study of the European Union. *Transforming Government: People, Process and Policy* 16, 3 (June 2022), 347–358. <https://doi.org/10.1108/TG-06-2021-0106>
- [82] Supadchaya Puangpontip and Rattikorn Hewett. 2020. Energy Usage of Deep Learning in Smart Cities. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 1143–1148.
- [83] Lucas Høyberg Puvus de Chavannes, Mads Guldborg Kjeldgaard Kongsbak, Timmie Rantza, and Leon Derczynski. 2021. Hyperparameter Power Impact in Transformer Language Model Training. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*. Association for Computational Linguistics, Virtual, 96–118. <https://doi.org/10.18653/v1/2021.sustainlp-1.12>
- [84] Crefeda Faviola Rodrigues, Graham Riley, and Mikel Luján. 2018. SyNERGY: An energy measurement and prediction framework for Convolutional Neural Networks on Jetson TX1. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*. The Steering Committee of The World Congress in Computer Science, 375–382.
- [85] Friederike Rohde, Maike Gossen, Josephin Wagner, and Tilman Santarius. 2021. Sustainability challenges of Artificial Intelligence and Policy Implications. *Ökologisches Wirtschaften-Fachzeitschrift* 36, O1 (2021), 36–40.
- [86] Bita Darvish Rouhani, Azalia Mirhoseini, and Farinaz Koushanfar. 2016. DeLight: Adding Energy Dimension To Deep Neural Networks. In *ISLPED '16: Proceedings of the 2016 International Symposium on Low Power Electronics and Design*. Association for Computing Machinery, New York, NY, USA, 112–117. <https://doi.org/10.1145/2934583.2934599>
- [87] Kanokwan Rungsuptaweepon, Vasaka Visoottiviseth, and Ryousei Takano. 2017. Evaluating the power efficiency of deep learning inference on embedded GPU systems. In *2017 2nd International Conference on Information Technology (INCIT)*. IEEE, 1–5.
- [88] Roy Schwartz, Jess Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commun. ACM* 63, 12 (Nov. 2020), 54–63. <https://doi.org/10.1145/3381831>
- [89] Ilia Shumailov, Yiren Zhao, Daniel Bates, Nicolas Papernot, Robert Mullins, and Ross Anderson. 2021. Sponge examples: Energy-latency attacks on neural networks. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 212–231.
- [90] Martino Sorbaro, Qian Liu, Massimo Bortone, and Sadique Sheik. 2020. Optimizing the energy consumption of spiking neural networks for neuromorphic applications. *Frontiers in neuroscience* 14 (2020), 662.
- [91] Dimitrios Stamoulis, Ermao Cai, Da-Cheng Juan, and Diana Marculescu. 2018. HyperPower: Power- and memory-constrained hyper-parameter optimization for neural networks. In *2018 Design, Automation, and Test in Europe Conference*. 19–24. <https://doi.org/10.23919/DATE.2018.8341973>
- [92] Dimitrios Stamoulis, Ting-Wu Rudy Chin, Anand Krishnan Prakash, Haocheng Fang, Sribhuvan Sajja, Mitchell Bognar, and Diana Marculescu. 2018. Designing adaptive neural networks for energy-constrained image classification. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. ACM, 1–8.
- [93] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 3645–3650. <https://doi.org/10.18653/v1/P19-1355>
- [94] Yuyang Sun, Zhixin Ou, Juan Chen, Xinxin Qi, Yifei Guo, Shunzhe Cai, and Xiaoming Yan. 2021. Evaluating Performance, Power and Energy of Deep Neural Networks on CPUs and GPUs. In *National Conference of Theoretical Computer Science*. Springer, 196–221.
- [95] Yuxuan Sun, Sheng Zhou, and Deniz Gündüz. 2020. Energy-aware analog aggregation for federated learning with redundant data. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 1–7.
- [96] Guglielmo Tamburini. 2022. The AI Carbon Footprint and Responsibilities of AI Scientists. *Philosophies* 7, 1 (Jan. 2022), 4. <https://doi.org/10.3390/philosophies7010004>
- [97] Yudong Tao, Rui Ma, Mei-Ling Shyu, and Shu-Ching Chen. 2020. Challenges in Energy-Efficient Deep Neural Network Training With FPGA. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- [98] Roberto Verdecchia, Luis Cruz, June Sallou, Michelle Lin, James Wickenden, and Estelle Hotellier. 2022. Data-Centric Green AI An Exploratory Empirical Study. In *2022 International Conference on ICT for Sustainability (ICT4S)*. IEEE, 35–45. <https://doi.org/10.1109/ICT4S55073.2022.00015>
- [99] Chengcheng Wan, Muhammad Santriaji, Eri Rogers, Henry Hoffmann, Michael Maire, and Shan Lu. 2020. {ALERT}: Accurate learning for energy and timeliness. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. 353–369.
- [100] Cong Wang, Bin Hu, and Hongyi Wu. 2022. Energy Minimization for Federated Asynchronous Learning on Battery-Powered Mobile Devices via Application Co-running. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. 939–949. <https://doi.org/10.1109/ICDCS54860.2022.00095>
- [101] Qu Wang, Yong Xiao, Huixiang Zhu, Zijian Sun, Yingyu Li, and Xiaohu Ge. 2021. Towards Energy-efficient Federated Edge Intelligence for IoT Networks. In *2021 IEEE 41st International Conference on Distributed Computing Systems Workshops*

- (ICDCSW). 55–62. <https://doi.org/10.1109/ICDCSW53096.2021.00016>
- [102] Yu Wang, Rong Ge, and Shuang Qiu. 2020. Energy-Aware DNN Graph Optimization. *Resource-Constrained Machine Learning (ReCoML) Workshop of MLSys 2020 Conference* (May 2020). <https://doi.org/10.48550/arXiv.2005.05837>
- [103] Yue Wang, Ziyu Jiang, Xiaohan Chen, Pengfei Xu, Yang Zhao, Yingyan Lin, and Zhangyang Wang. 2019. E2-Train: Training State-of-the-art CNNs with Over 80% Energy Savings. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2019/file/663772ea088360f95bac3dc7fb841be-Paper.pdf>
- [104] Simon Wenninger, Can Kaymakci, Christian Wiethe, Jörg Römmelt, Lukas Baur, Björn Höckel, and Alexander Sauer. 2022. How Sustainable is Machine Learning in Energy Applications? – The Sustainable Machine Learning Balance Sheet. [https://aisel.aisnet.org/wi2022/sustainable\\_it/sustainable\\_it/1](https://aisel.aisnet.org/wi2022/sustainable_it/sustainable_it/1)
- [105] Martina Willenbacher, Torsten Hornauer, and Volker Wohlgemuth. 2021. Rebound Effects in Methods of Artificial Intelligence. In *Advances and New Trends in Environmental Informatics*. Springer, Cham, Switzerland, 73–85. [https://doi.org/10.1007/978-3-030-88063-7\\_5](https://doi.org/10.1007/978-3-030-88063-7_5)
- [106] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable AI: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems* 4 (2022), 795–813.
- [107] Haichuan Yang, Yuhao Zhu, and Ji Liu. 2019. Energy-constrained compression for deep neural networks via weighted sparse projection and layer input masking. *International Conference on Learning Representations (ICLR)* (2019).
- [108] Tien-Ju Yang, Yu-Hsin Chen, and Vivienne Sze. 2017. Designing Energy-Efficient Convolutional Neural Networks Using Energy-Aware Pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [109] Xiangyu Yang, Sheng Hua, Yuanming Shi, Hao Wang, Jun Zhang, and Khaled B. Letaief. 2020. Sparse Optimization for Green Edge AI Inference. *Journal of Communications and Information Networks* 5, 1 (2020), 1–15. <https://doi.org/10.23919/JCIN.2020.9055106>
- [110] Zhaohui Yang, Mingzhe Chen, Walid Saad, Choong Seon Hong, and Mohammad Shikh-Bahaei. 2020. Energy efficient federated learning over wireless communication networks. *IEEE Transactions on Wireless Communications* 20, 3 (2020), 1935–1949.
- [111] Chunrong Yao, Wantao Liu, Weiqing Tang, Jinrong Guo, Songlin Hu, Yijun Lu, and Wei Jiang. 2021. Evaluating and analyzing the energy efficiency of CNN inference on high-performance GPU. *Concurrency and Computation: Practice and Experience* 33, 6 (2021), e6064.
- [112] Barzan A Yosuf, Sanaa H Mohamed, Mohammed M Alenazi, Taisir EH El-Gorashi, and Jaafar MH Elmirmighani. 2021. Energy-Efficient AI over a Virtualized Cloud Fog Network. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*. 328–334.
- [113] Jia-Ruei Yu, Chun-Hsien Chen, Tsung-Wei Huang, Jang-Jih Lu, Chia-Ru Chung, Ting-Wei Lin, Min-Hsien Wu, Yi-Ju Tseng, and Hsin-Yao Wang. 2022. Energy Efficiency of Inference Algorithms for Clinical Laboratory Data Sets: Green Artificial Intelligence Study. *J. Med. Internet Res.* 24, 1 (Jan. 2022), e28036. <https://doi.org/10.2196/28036>
- [114] Boyu Zhang, Azadeh Davoodi, and Yu Hen Hu. 2018. Exploring Energy and Accuracy Tradeoff in Structure Simplification of Trained Deep Neural Networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 8, 4 (2018), 836–848. <https://doi.org/10.1109/JETCAS.2018.2833383>
- [115] Xingzhou Zhang, Yifan Wang, and Weisong Shi. 2018. pCAMP: Performance Comparison of Machine Learning Packages on the Edges. In *USENIX workshop on hot topics in edge computing (HotEdge 18)*.
- [116] Sha Zhu, Kaoru Ota, and Mianxiong Dong. 2021. Green AI for IIoT: Energy Efficient Intelligent Edge Computing for Industrial Internet of Things. *IEEE Trans. Green Commun. Networking* 6, 1 (Aug. 2021), 79–88. <https://doi.org/10.1109/TGCN.2021.3100622>

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/369014627>

# AI4CE -Closing the design-operation-loop with AI: design, operate, learn, repeat

Conference Paper · March 2023

---

CITATIONS

0

READS

45

4 authors, including:



Jan-Peter Ceglarek  
Technische Universität Darmstadt

1 PUBLICATION 0 CITATIONS

[SEE PROFILE](#)



Redouane Boumghar  
Parametry.ai

23 PUBLICATIONS 109 CITATIONS

[SEE PROFILE](#)



Noel Png  
1 PUBLICATION 0 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Project OPSWEB [View project](#)



Project AI for Space Operations [View project](#)

## AI4CE - Closing the design-operation-loop with AI: design, operate, learn, repeat

Jan-Peter Ceglarek <sup>a\*</sup>, Redouane Boumghar<sup>b</sup>, Noel Jiahao Png <sup>c</sup>

<sup>a</sup> Technical University Darmstadt, Parametry.ai, [ceglarek@fsr.tu-darmstadt.de](mailto:ceglarek@fsr.tu-darmstadt.de)

<sup>b</sup> Parametry.ai, [red@parametry.ai](mailto:red@parametry.ai)

<sup>c</sup> Parametry.ai, [noel@parametry.ai](mailto:noel@parametry.ai)

\* Corresponding Author

### Abstract

Modern system designs has to take a lot of requirements into account, but real-world experience is rarely given the opportunity to have a direct impact on new mission designs. AI for Concurrent Engineering (AI4CE) is a open source research project between TU Darmstadt and Parametry.ai, which examines the applicability of AI-based system generation to support the conceptual design phase. As one of several modules, OPS2Design will offer a formalised way to introduce constraints and requirements from the operations directly into the generation of new system designs, thereby closing the PLC loop.

**Keywords:** AI, system design, conceptual design, concurrent engineering, operational experience

## 1. Introduction

The design of a space mission is influenced by many factors, from the engineering details to finance, risk and scientific aspects. To accelerate the decision process and increase its efficiency, concurrent engineering (CE) has proven itself to be a valuable tool for agencies and industry alike. Research to further optimise the CE process identified modern AI tools of being able to further provide valuable advantages [1] [2] Past efforts to provide AI-based design expertise can be divided into two categories of approaches: Capable AI agents can offer additional design knowledge, by offering deeper insights about the experiences and decisions taken in previous design studies via text-based interfaces [ 1][2] (top-down) or by providing new, unbiased designs generated by AI-powered system generation tools (bottom-up). These two approaches both aim to support the CE teams in their pursue to develop conceptual space systems efficiently. They do differ, however, in the way they generate the supporting design knowledge - the first one extracting existing design wisdom to set it into a novel context, the latter one generating entirely new, design expertise just based on requirements and established engineering calculations. Additionally, they differ significantly in the scale of research conducted to date, with a clear focus on the top down method.

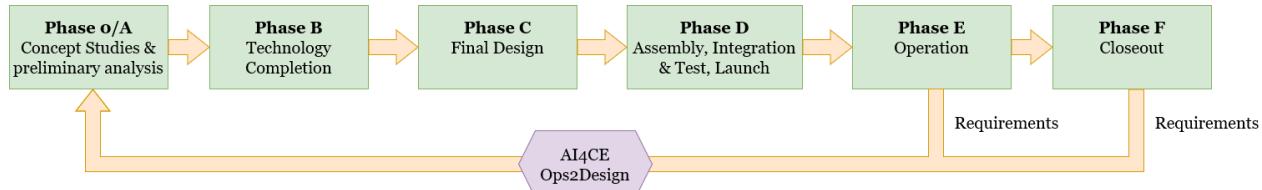
To generate space systems from the bottom up, the AI selects a set of suitable components out of a database of available components, based on a set of mission requirements. These requirements are provided in a format, which can be imported by the generating AI. This applies both to the requirements imposed on the system by the customer or the designing team, and to any requests that the operation may formulate for the basic design of the system.

The Artificial Intelligence for concurrent engineering (AI4CE) research project, as a collaboration between the TU Darmstadt and Parametry.ai, explores the use of AI-assisted system generation to support early design development. The core of the research is a platform to enable and support the development as well as validation of bottom up system generation

## 2. Bottom-up AI-assisted System Development

Modern system development uses Model Based System Engineering (MBSE), which describes a procedural product development process, where all information from the first development phases of the system idea and in all later phases of the design, get collected in one single source of truth. It is a constant effort to unify all relevant aspects of the space industry into one model, which would facilitate the model exchange across agency or company borders. While this is mainly done for the early phases of the design phase, adding this approach to the whole Product Life Cycle (PLC) - including operations - can add additional possibilities to learn from it in succeeding design studies. The common, linear PLC of a spacecraft starts in Phase 0/A with its first concept studies, gets developed, refined, tested, launched and used during Phases B-E, until it gets finally deposed during Phase F. Automatically linking the experience about specific components and component combinations during operations (Phase E) back to the start of

the mission design (Phase 0/A) - thereby closing the PLC loop, as shown in Fig.1 - will provide an extra level of knowledge to the designing team previously unknown.



**Figure 1** Each mission traverses through the same Product Life Cycle (PLC) [3]. AI4CE will close the PLC loop, by connecting Phase E/F of the current to Phase 0/A of the next mission, through a formalised way to describe requirements, which can be considered by the automated system generation.

The automated generation of conceptual designs can benefit from the operational experience in many ways and the advent of the digital twin concept made the implementation of experiences of operating previous spacecraft designs increasingly feasible. Mass production of satellite components caused, that satellite components get used in multiple missions. This means, that actual real world knowledge about operating these components can be gathered from the operation teams. Due to the long development and construction time of space systems, it is difficult to perceive a concrete impact of the design decisions. Therefore, it is extremely beneficial to get a direct influence of the operation during the creation of the design from operational experience. This is currently done by interviewing operation teams and including them into CE sessions. The design of a system is constraint by its requirements, which in AI4CE will be extended to not only comply to the mission objective, but also to cover the needs of an efficient operation. Modern space missions can chose from a variety of space components and have to rely on their own user experience and the data sheets of the vendors. Although it is not the dedicated subject of the preliminary design studies in Phase 0/A to fully design a system, an understanding of what real world components could be used to build the current version of the envisioned design will be of great value during the CE sessions.

### 3. Theory and calculation

The core of AI4CE is a digital platform for the design, implementation, test and validation of different system generation methods to support the CE process. The goal is to enable the design of AI-based system generators, which are tightly integrated into the CE workflow and require thereby a high level of integration with the designing engineers. The outlined operational workflow of a bottom-up system generation is visualised in Figure 2. The main focus is on the intensive provision of bottom-up procedures, through the systematic development of generation and validation tools. However, the platform is designed in a modular way, so that top-down procedures can also be taken into account by future studies. The bottom-up system generation process relies on the input of mission requirements together with databases of available components for the artificially generates mission designs. The input requirements for the system consist of the general technical requirements for a space system (environmental influences, remote operations), the specific requirements of the mission objectives (scientific instruments, commercial services, ...) and the requirements addressed from the experience of operational use. The team will update the mission requirement parameters according to their progressed design discussion, on which a novel concept can get generated by the AI system creator. To ensure a seamless integration into existing MBSE-centric CE activities and other MBSE tools, a dedicated MBSE integration module (DCC2Design) will be part of the AI4CE platform. Part of this module will be the functionality to formalise system requirements

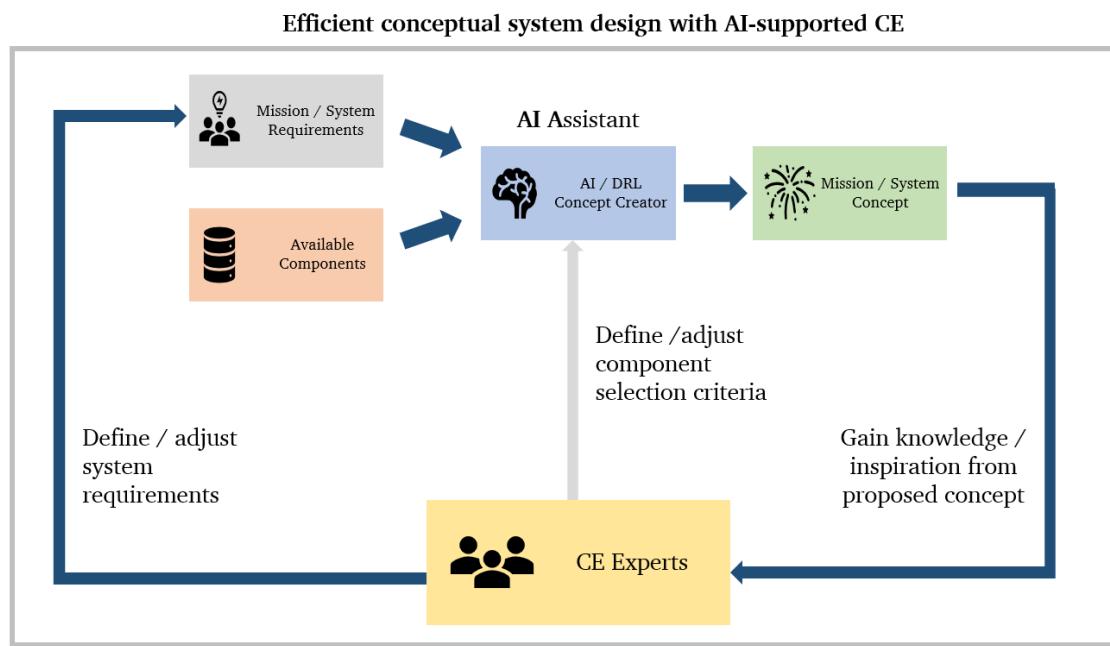
### 4. Results

In order to bridge the gap from to operations to the design of a mission, multiple prerequisite research questions (RQ) are yet to be answered.

RQ1. Which metrics can be used to identify the relevant criteria of ongoing and historical space missions which have the most relevant impact on the next generation of spacecrafts? To measure the impact of a space mission on the design of the next generation, first the relevant criteria have to be identified. While in classical mission design parameters like mass or power consumption are key design constraints, operations requires its own set of prerequisites. Expert interviews with operators of different domains will help identify their dedicated needs

RQ2. How can these requirements be formalised in a way, that they can be used for the Phase 0/A studies? Once the mission (design) critical parameters have been identified they can be fixed or monitored, but that still does not mean, that they can be usefully implemented during the mission design phase. The corresponding requirements have to be formalised in a schema, that can be implemented during Phase 0/A studies.

RQ3. How can the influence be introduced to the design of a new mission? Once the necessary aspects of the space missions have been identified, formalised and collected, they still need to be implemented into the actual mission design. In regular mission designs this is handled as a set of requirements, which got defined by the mission's customer before the concept study and evaluated with the engineering team after the study concluded. Therefore once the requirements are consistent with the fundamental requirements coming from the mission's initial customer, they can be implemented as regular requirements



**Figure 2** Overview of how a bottom up system concept creation workflow. The CE team receives information about the system's purpose and an MBSE model. Building on that, the team adjust the AI system's mission requirements. Together with additional information coming from the operations and a component database, a concept gets created. The CE experts will then review the created system and start a new iteration, until the final design is reached, which gets exported as an updated MBSE model.

The three models of the AI4CE project are highlighted as purple boxes.

## 5. Discussion

AI4CE is currently still in its early phase. A first prototype showcasing the feasibility of bottom-up system generation uses Deep Reinforcement Learning (DRL) to learn the best component combinations from the satsearch web shop. The generator is currently capable of generating real-world comparable simplified 1U CubeSat systems with support for solar panel, battery, camera, reaction wheel and transceiver module. The two other modules of AI4CE are the Deep Reinforcement Learning (DRL) Concept Creator (DCC) and OPS2Design, which functions as an integration module of component experience/knowledge during the actual operation phase. All 3 modules are marked purple in Figure 2. In addition to the general functionality of the bottom-up system generation process outlined above, the AI4CE platform will offer support tools and functionality to develop these bottom-up system generation tools. It is the selected goal to offer functionality to compare system generation methods. A GUI builder will help to outline the envisioned system architecture, which will be used by the DCC module to generate the system. After the system architecture has been configured, details about the be-wished AI method can be adjusted, before the system generation can be started. For validation of the generated system, a comparison module is

envisioned to benchmark different system generation methods against each other. This will be achieved by special metrics to compare the outcome of numerous AI algorithms as well as static optimisation algorithms and designs created by human design teams. On behalf of the integration into a greater ecosystem of an industry-wide used tools for system design, AI4CE will also offer the functionality to export the generated design in other MBSE tools by relying on open system exchange formats, like explored by ESA’s OSMOSE initiative.

## 6. Conclusions

Space operations and mission design are aspects of a spacecraft’s PLC which mark the beginning and the end. Both procedures require dedicated, experienced personnel with special training and an exchange of operational experience to support the design process. AI4CE is a research project in cooperation between parametry.ai and the Technical University Darmstadt, which will provide a platform to build and validate AI-based engineering assistants to support CE studies. One focus of the research is on formalising design requirements so that an automated design tool can generate systems based on them, which means that design generation is also strongly based on requirements derived from operational experience. Although the development is still in its early steps, a first prototype offered promising results. Next steps in the development process will focus on the identification of relevant operational criteria that will influence the mission design and a schema for the formalisation of requirements. The project is completely open source and is licenced under the GPL3 license <https://gitlab.com/ai4ce/public-info>

## References

- [1] Berquand, A., Murdaca, F., Riccardi, A., Soares, T., Generé, S., Brauer, N., and Kumar, K., “Artificial intelligence for the early design phases of space missions,” *2019 IEEE Aerospace Conference*, IEEE, 2019, pp. 1–20.
- [2] Fleith, P., “AstroSQuAD: Building blocks for the development of an Astronautics & Space Question-Answering Dataset to benchmark machine comprehension of text,” *IAC\_2022*, ????
- [3] “Basics of space flight section II. space flight projects,” , ????, URL <https://www2.jpl.nasa.gov/basics-bsf7-1.php>.



# Speech emotion recognition with artificial intelligence for contact tracing in the COVID-19 pandemic

Francesco Pucci<sup>1,2</sup> | Pasquale Fedele<sup>2</sup> | Giovanna Maria Dimitri<sup>1</sup>

<sup>1</sup>DIISM, Università degli Studi di Siena, Siena, Italy

<sup>2</sup>Blu Pantheon, Siena, Italy

## Correspondence

Giovanna Maria Dimitri

Email: [giovanna.dimitri@unisi.it](mailto:giovanna.dimitri@unisi.it)

## Abstract

If understanding sentiments is already a difficult task in human-human communication, this becomes extremely challenging when a human-computer interaction happens, as for instance in chatbot conversations. In this work, a machine learning neural network-based Speech Emotion Recognition system is presented to perform emotion detection in a chatbot virtual assistant whose task was to perform contact tracing during the COVID-19 pandemic. The system was tested on a novel dataset of audio samples, provided by the company Blu Pantheon, which developed virtual agents capable of autonomously performing contacts tracing for individuals positive to COVID-19. The dataset provided was unlabelled for the emotions associated to the conversations. Therefore, the work was structured using a sort of transfer learning strategy. First, the model is trained using the labelled and publicly available Italian-language dataset EMOVO Corpus. The accuracy achieved in testing phase reached 92%. To the best of their knowledge, this work represents the first example in the context of chatbot speech emotion recognition for contact tracing, shedding lights towards the importance of the use of such techniques in virtual assistants and chatbot conversational contexts for psychological human status assessment. The code of this work was publicly released at: <https://github.com/fp1acm8/SER>.

## KEY WORDS

affective computing, artificial intelligence, machine learning

## 1 | INTRODUCTION

Over the past 2 years, since humanity has been dramatically affected by COVID-19, our lives have drastically changed. COVID-19, acronym of Corona Virus Disease 19 [1], was first identified in Wuhan, China, in December 2019 and became a pandemic in 2020 [2].

Among the many challenges that the health system was facing during the pandemic, there was the crucial issue of contact tracing. By contact tracing we intend the search and contact management of a confirmed COVID-19 case. This was an essential public health action to pursue, to try to limit the ongoing epidemic. Indeed, identifying and managing the contacts of confirmed COVID-19 cases allowed to quickly identify and isolate any secondary cases, thus interrupting the infection transmission chain [3].

The operation of contact tracing required the employment of several human resources. In particular, this task affected dramatically the local Italian health authority (the Italian so called ASLs). Therefore an innovative approach to offer a solution to this issue consisted in the use of conversational interfaces based on artificial intelligence (AI) and Natural Language Processing (NLP).

The importance of emotion recognition in contact tracing dialogues is manifold. For instance it could help companies in detecting potential stress and psychological conditions of people involved in the conversations with the chatbots, as well as detecting potential liars. Furthermore, using the information obtained, the companies developing chatbots could decide to modify the flow of dialogues. Last, but not least, when worrying psychological emotions were to be detected by the automatic system, the company could decide to proceed

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2023 The Authors. *Cognitive Computation and Systems* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Shenzhen University.

further and make the human talk to a psychologist to help the person in need of further psychological help.

The startup company Blu Pantheon [4], with whom we collaborated, works in the development of virtual assistants. In particular, in our specific case, the goal of our work consisted in the implementation of a neural network-based system, capable of accurately identifying emotions starting from speeches. The speech emotion recognition model we implemented was tested on a novel dataset provided by Blu Pantheon. In particular, such dataset was collected thanks to a virtual contact tracing chatbot, developed by Blu Pantheon, which allowed to perform contact tracing during the first wave of the pandemic in Italy.

In our paper, we presented several novelties. First of all, to the best of our knowledge, we introduced, for the first time in the literature, the research question related to understanding emotions from contact tracing conversation in a chatbot context, with particular reference to the COVID-19 contact tracing task. Secondly, we used a completely novel dataset, collected by the startup Blu Pantheon. Thirdly, we used a transfer learning approach by using a dataset built for a similar but different context, characterised by the same language (i.e. Italian).

The paper is organised as follows: In Section 2, we report a comprehensive literature review for speech emotion recognition (SER) in AI.

In particular, we focussed on the role of SER for cognitive and psychological assessment of the health status of users. In Section 3, we report a comprehensive description of the datasets we used and of all of the pre-processing steps we performed. In Section 4, we describe the computational workflow and the methodologies used in our experiments. In Section 5, we described the experimental settings. In Section 6, we described the conclusions deriving from our work, together with a thorough description of the possible future developments.

## 2 | LITERATURE REVIEW AND BACKGROUND

In the latest years, machine learning (ML) and artificial intelligence (AI) has been successfully applied to many different fields [5–9]. Moreover, throughout the last 2 decades, research focussed on automatic emotions recognition using ML have been developed [10]. In the context of contact tracing, however, only a few works can be found, which relates emotion recognition to contact tracing conversations.

In Ref. [11], the authors analysed user reviews collected from the Irish Health Service Executive's (HSE) Contact Tracker app. The app was developed with the aim of identifying large-scale and automated analysis of reviews. A total of 1287 reviews from the Google/Apple playstores was collected to classify aspects of the app on which the users mostly focussed.

To the best of our knowledge, no works are present in the literature for what concerns the Italian language, and for non-

app tracing related contexts, showing the complete novelty of the framework and analysis proposed in our work.

In the following subsections, we will present an overview of the main background concepts that are necessary to better understand the overall context in which our paper is focussed.

In Section 2.1, we will summarise concepts related to the field of Emotion AI. In Section 2.2, we will report a comprehensive summary of the available databases and corpora for emotion recognition. In Section 2.3, we will report a comprehensive overview of Speech Emotion Recognition Systems and the related ML approaches. In Section 2.4, we will present an overview of ML and related SER applications.

### 2.1 | Emotion AI

In a world where technology advances at exponential speed, Human-Computer Interaction (HCI) has become one of the most studied fields of research. The goal of HCI is not only to create a communicative interface that is as natural as possible between human and machine but it is also to create new communication paradigms that can improve human life. Emotions are, in fact, one of the predominant aspects of human interactions and, consequently, they have also become an important aspect of the development of HCI-based applications.

Emotions can be technologically captured and assessed in a variety of ways, such as facial expressions, physiological signals or speech. With the intention of creating more natural and intuitive communication between humans and computers, emotions conveyed through signals should be correctly detected and appropriately processed. Throughout the last 2 decades of research focussed on automatic emotions recognition, several machine learning techniques have been developed and constantly improved. This field of research is widely known as Emotion AI. Emotion AI can have several applications:

- Video gaming.** Using computer vision, the game console detects emotions via facial expressions during the game session and adapts to it [12]. This is often performed during the testing phase of a video game.
- Healthcare.** Several applications can be found in the healthcare sector. For example, voice analysis software can help doctors with the diagnosis of diseases such as depression and dementia. Another application is the use of ‘nurse bot’ not only to remind older patients on long-term medical programmes to take their medication but also talk with them every day to monitor their overall well-being [13].
- Education.** Learning software prototypes have been developed to adapt to kids’ emotions. When the child shows frustration because a task is too difficult or too simple, the programme adapts the task so it becomes less or more challenging. Another learning system helps autistic children recognise other people’s emotions [14].
- Employee safety.** Based on Gartner client inquiries<sup>1</sup>, demand for employee safety solutions is on the rise. Emotion AI can help, for instance, to analyse the stress

- and anxiety levels of employees who have very demanding jobs such as first responders.
5. **Automotive sector.** Automotive vendors can use computer vision technology to monitor the driver's emotional state. An extreme emotional state or drowsiness could trigger an alert for the driver. Another possible application is in the future of autonomous vehicles, where many sensors such as cameras and microphones could help to monitor what is happening and understand how users view the driving experience [15].
  6. **Fraud detection.** Insurance companies use voice analysis to detect whether a customer is telling the truth when submitting a claim. According to independent surveys, up to 30% of users have admitted to lying to their car insurance company in order to gain coverage.
  7. **Recruiting.** The software could be used during job interviews to understand the credibility of a candidate.
  8. **Call centre intelligent routing.** An angry customer can be detected from the beginning and can be routed to a well-trained agent who can also monitor in real-time how the conversation is going and adjust [16].
  9. **Connected home.** A VPA-enabled speaker (e.g. Google Home, Alexa, etc.) can recognise the mood of the person interacting with it and respond accordingly. Recent advances in Amazon's quest for humour detection are worth mentioning [17].
  10. **Public service.** Partnerships between emotion AI technology vendors and surveillance camera providers have emerged. Cameras in public places in the United Arab Emirates can detect people's facial expressions and, hence, understand the general mood of the population. This project was initiated by the country's Ministry of Happiness.
  11. **Retail.** Retailers have started looking into installing computer vision emotion AI technology in stores to capture demographic information and visitors' mood and reactions.

## 2.2 | Emotional speech databases or corpora

Over the last few years, the development of SER systems has led to the creation of labelled databases for emotion recognition.

Most of them are freely available and can be retrieved online. The databases used for the development of the SER system differ in language, number of different speakers, emotions represented and finally the type. The three types highlighted in Ref. [18] are simulated, natural and induced. We will now briefly describe all of the three types.

### 2.2.1 | Simulated emotional speech databases

A simulated/acted emotional speech database is a type of database with audio samples collected from actors who are aware of the recording process [18].

The recording process consists of a list of sentences and a certain set of emotions. Actors are asked to simulate a sentence for each emotional state. The advantage of this type of database collection is that researchers get full control of the quality of the recordings and therefore it is also easier to construct.

However, the disadvantage is that the natural component is neglected, and the resulting model may not work in real-time emotion recognition application.

### 2.2.2 | Natural emotional speech databases

A natural emotional speech database is a type of database with audio samples recorded in a natural environment. In general, it consists of recordings taken from call centres conversations, talk shows or movies [18]. In this case, speakers are not aware of the recording process. The advantage of this type of database is its reliability, since emotions are natural and the resulting SER model may obtain much better accuracy in real-time emotion recognition applications.

The disadvantage is the complexity in constructing and analysing it because emotions in everyday life may be less expressive and therefore recognisable compared to acted emotions. Another drawback is that it often contains unbalanced emotional categories.

### 2.2.3 | Induced emotional speech databases

An induced/elicted emotional speech database is a type of database with audio samples recorded under simulated natural conditions.

This means that speakers are put into situations to induce a specific emotion. That is why these types of databases are considered more natural than simulated databases but they are not entirely natural [18].

### 2.2.4 | Examples of existing emotional speech databases

Other emotional speech databases can be derived from multimodal emotional databases which include textual and visual data in addition to audio samples. Some of the best known in the English language are Crowd-sourced Emotional Multimodal Actors Dataset (Crema-D) [19], Ryerson Audio-Visual Database of Emotional Speech and Song (Ravdess) [20], Surrey Audio-Visual Expressed Emotion (Savee) [21] and Toronto emotional speech set (Tess) [22]. Another notable example is the IEMOCAP database [23]. IEMOCAP stands for 'Interactive Emotional Dyadic Motion Capture Database.' It consists of a corpus collected by the University of Southern California (USC) in which 10 actors were recorded while keeping sensors on faces, heads and hands. It represents an extremely important database for sentiment analysis. Also, in this case, the recorded speech are in English.

## 2.3 | Speech emotion recognition and machine learning

ML methodologies have, nowadays, reached the state of the art performances in several different fields: from bioinformatics to computer vision, from anomaly detection to computer forensic [5, 24–29].

The capability of machine learning to automatically perform difficult prediction and classification tasks has allowed to move steps forwards in solving tasks which were unsolvable a few years ago. In this context, new advances in research stands for emotion detection and recognition.

Emotions can be technologically collected in different ways, and multiple approaches can be used for their recognition. One example is the multimodal approach which aims to combine textual, audio and visual data [30].

However, in the present work, only audio data was used, as the novel dataset we analysed was audio only. In this case, we are no longer referring to Emotion Recognition in general but to the field of SER. This can be defined as extraction of the speaker emotional state from the speech signal [31]. Input data for SER systems consist of audio signals that are analogue representations of a sound.

Several approaches have been proposed in the literature, trying to exploit different features sound. The approaches used range from classic feature extraction to the implementation of different types of classifiers (i.e., Gaussian mixture model, Hidden Markov model, Support Vector Machine, Artificial Neural Network, etc.) [32] as well as deep learning models that act directly on sound representations such as the spectrogram and the time series (i.e., waveform). These latter allow automatic feature extractions, however requiring the availability of big datasets for model training [33]. For instance in Ref. [34], the authors use a Hidden Markov Model (HMM) approach to predict emotions. They obtained an accuracy of 80% in recognising seven different emotional states (disgust, fear, anger, joy, surprise, sadness and neutral) using the best combination of low level sound features (pitch and energy).

In Ref. [35], instead, the authors use a Support Vector Machine to classify five different emotional states (disgust, boredom, sadness, neutral, and happiness). They obtained 66.02% classification accuracy only by using energy and pitch features and 70.7% by using exclusively Linear Prediction coefficients and Mel cepstrum coefficients (LPCMCC) features extracted from the audio files and obtaining an accuracy of 82.5% using both of them.

Moreover in Ref. [36], the authors use a Multilayer Perceptron to recognise four emotional states (happy, angry, sad and neutral) with an overall accuracy of 81%. The network used is composed of an input layer, one hidden layer and the output of the four classes. Features extraction considered both temporal and spectral features for the classification task. The latest development of ML, that is, deep learning has also reached the state of the art for what concerns emotion recognition and classification. In particular, Convolutional Neural Networks (CNNs) have often reached state-of-the-art performances. For instance in Ref. [37], the authors used

CNN-based architectures to perform SER on unlabelled samples. Performances were assessed on four public databases. A further example is, for instance, [38] where a real-time SER system based on dilated convolutions was proposed (DCNN). Residual blocks were proposed in order to learn the long-term contextual dependencies in the input features, and the features were later concatenated to perform the final emotion tasks. The architecture was tested on the IEMOCAP and EMO-DB benchmark datasets, obtaining a high recognition accuracy of 73% and 90% for each of the benchmarks. Among the most recent approaches, we can find the Wav2Vec2 approach, which recently reached the state-of-the-art performances for what concerns speech emotion recognition [39]. In particular, in Ref. [39], the author proposed to mask the speech into the latent space and use a contrastive loss so that the speech input is learnt. The method has now reached state-of-the-art performances in several different fields. Another state-of-the-art model in SER, which is worth mentioning, is in Ref. [40]. The proposed Hidden-Unit BERT (HuBERT) models aim at approaching the self-supervised speech representation learning field. The idea is to use an offline clustering step, in order to provide aligned target labels, with the final result of having a BERT-like prediction loss. The computational pipeline implemented includes a CNN and transformer encoding part together with a K-Means approach, resulting in a final model that is able to improve performances of BERT.

Deep learning (DL) methods (mainly based on CNN) have the great advantage of not having to specify the features of the sound to be used in advance. In this way, the extraction step, which could include human biases as well as a time consuming approach, is removed from the pipeline. However the amount of labelled data and, in general, the dimensionality of the dataset, which can be used in this case, is relevant, and therefore DL methods cannot be used in small unlabelled dataset as the case of the novel dataset we analysed and presented in our work.

## 2.4 | Speech emotion recognition (SER) and COVID-19

In this section, we will introduce literature related to the case of Speech Emotion Recognition for COVID-19-related research. COVID-19 has in fact significantly affected our lives and our emotions. The virus highly affected our way of interacting and significantly affected emotional and facial recognition in human interaction. Several researches have in fact investigated the relationship existing between facial emotion recognition, COVID-19 and emotions recognition [41]. In this context, several studies have in fact focussed on the possibility of identifying emotions from masked faces [41–43].

For what concerns speech emotion recognition and text, however, not the same can be said. A few studies, in fact, can be found, which relate emotion recognition and COVID-19-related texts. For instance, in Ref. [44], the authors analysed emotions from the data obtained from the TraceTogether app

and conducted a cross-sectional survey at the large public hospital in Singapore after the COVID-19 lockdown.

Moreover in Ref. [45], the authors used Twitter-based analysis for understanding people's feelings on social distancing from Twitter's data.

The data streams were analysed through the use of a Deep Learning approach (Deep Belief Networks) with pseudo-labelling. Moreover in Ref. [46], the authors analysed over 500.000 tweets related to COVID-19 from UK cities (collected in the last 2 years from February 2020 to November 2021). Using different types of deep learning approaches (based on emotion recognition and topic modelling), it was possible to observe the difference in sentiments related to vaccination and epidemiological situation. Moreover in Ref. [47], the authors collected more than 2 millions tweets in the period from February-June. In this way, a multi-class classifier was trained and used for understanding the Twitter COVID-19-related sentiments, achieving a classification accuracy of 80.33%. In all of the research cases described so far, the text features were extracted from the Tweets by using several different deep learning models and allowing to therefore use them as input to different types of classifiers.

For what concerns speech emotion recognition and COVID-19 related emotions, to the best of our knowledge, no works can be found in the literature in which the COVID-19 sentiment analysis was performed by using speech audio or text derived from audio. In this sense, our work represents an absolute novelty in this context and in the context of human-computer interaction sentiment analysis dialogues systems.

### 3 | MATERIALS

#### 3.1 | Blu Pantheon contract tracing system and dataset

Blu Pantheon [4] is a startup active in the process innovation market through the implementation of innovative solutions for telemedicine and computer vision, with a strong focus on the development of the aforementioned conversational interfaces based on AI and NLP. It is precisely through the use of such technologies that it proposes new application scenarios that use AI in sectors such as healthcare and smart cities.

The product of the company is the result of the activity of the R&D team led by Dr. Pasquale Fedele, a computer engineer and serial entrepreneur who received the award Knight Order of Merit of the Italian Republic in 2017 for having created with

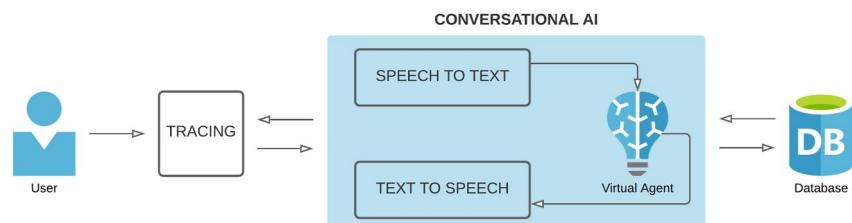
his other company LiquidWeb, a medical device (i.e. Brain-control AAC) that gives people affected by amyotrophic lateral sclerosis (ALS) the possibility to control assistive technologies through their thoughts [48].

We will now here briefly describe the tool designed by the company to help the ASLs in the contact tracing process. The proposed solution involves the use of a conversational AI system, capable of discriminating the information useful for contact tracing thanks to the syntactic/semantic analysis of the interaction flow. In the case of the identification of the list of names and related contacts communicated by the positive patient to COVID-19, the conversational voice-bot processes the speech through a Speech-to-Text module. Subsequently, the agent analyses the text, extrapolating the words that can be mapped back to names, surnames and telephone numbers. Once identified, it proceeds to store them in a specific database.

The virtual learning agent process takes place both through the supervision of specialised technicians and through self-learning through the application of machine learning algorithms. This module interacts with the ASL telephone platform, as well as with the customer's database. A summary of the architecture is shown in Figure 1. More precisely, a call consists of a set of questions, following a path derived on the base of the answers given by the user.

The questions can be divided into seven types:

- **Person identification.** The identity of the user is identified. For privacy reasons, the conversations relating to this section are not reported in the dataset provided to us.
- **Clinical interview.** In this step the questions are asked that concern the patient's symptoms (symptomatic or asymptomatic). Based on this, there are different procedures to follow according to the Italian law.
- **Tracing.** The patient is asked if he/she lives with other people and if he/she can provide their names, surnames, dates of birth and telephone numbers.
- **Cohabitant interview.** The name, surname, date of birth and telephone number of each cohabitant are stored.
- **Job investigation.** A person is asked if she/he is working, the last time he went to work, the name and telephone number of the company. Also this information for privacy reasons has not been provided in our novel dataset.
- **Social interview.** The user is asked if he/she had other contacts outside his/her cohabitants and work. If so, all contact information is required (name, surname, date of birth and telephone number).



**FIGURE 1** Blu Pantheon contact tracing service architecture.

- *Quarantine guidelines and Greetings.* At the end, a series of recommendations to follow are listed, and the user is asked if he wants to listen to them again before the phone call ends.

It is worth noticing that during the call, a section can be skipped depending on the answers given by the user. Furthermore, if an answer is not understood, the user is asked to repeat up to a maximum of three times. Subsequently, the call is dropped.

The service has been operating on an experimental basis since November 2020 in support of some ASLs in the Veneto Italian region, finding a context with various problems to be solved such as epidemiological investigation, neophyte personnel to be trained or unpredictability of the COVID-19 pandemic and consequent frequent regulatory adjustments. During such validation period, a single human operator, supervising a virtual agent, has been able to carry out on average 9.6 cases per hour for an estimated management cost of 28.26 €/h, while to carry out the same workload in the traditional way (i.e., proceeding with a human operator to make contact tracing calls and manually fill the database) requires seven human operators for an estimated cost of 91.96 €/h.

Moreover, this service allows ASLs to manage phone calls peaks more easily and allows healthcare personnel to save time that is useful to carry out tasks that require greater competence. Furthermore, the modularity of the service allows its use on multiple channels, for example, through text messages or voice interaction. This favours social inclusion towards people with disabilities such as deaf-mute, visual impairment or motor difficulties in the upper limbs.

The service has been operating on an experimental basis since November 2020 in support of some ASLs in the Veneto region, finding a context with various problems to be solved such as

- Epidemiological investigation designed for small numbers and transferred to very large numbers in a short time
- Neophyte personnel to be trained and standardised
- IT tools not suitable for large-scale investigations
- Databases and applications not in communication with each other
- Unpredictability of the COVID-19 pandemic and consequent frequent regulatory adjustments

The novel dataset provided by Blu Pantheon for our experiments consists of 3005 audio samples in WAV<sup>1,2</sup> format, with a sampling frequency of 48 kHz. The sampling frequency of 48 kHz was chosen in accordance to the sampling frequency of the EMOVO dataset. In this way, uniforming the one of EMOVO and the one of our novel dataset, we could proceed

training the dataset on EMOVO and then testing in the novel Blue Pantheon dataset.

The Blu Pantheon dataset was obtained from the recordings of the conversations between users and the virtual agent during the contact tracing calls. Each audio file corresponds to a single user response to a specific question of the virtual agent. For example, suppose the agent asks if the user had close contacts outside the family environment and the user replies ‘yes,’ then the audio file will consist of the recording of the user pronouncing the word ‘yes.’ The dataset provided is totally novel in which no descriptive label of the emotional state of the users is reported.

### 3.2 | EMOVO dataset

EMOVO is the first publicly available emotional corpus for the Italian language [49].

It is a database built using the voices of six actors (three males and three females) who played 14 sentences simulating six emotional states (disgust, fear, anger, joy, surprise and sadness) plus the neutral state [49]. These emotions are the well-known Big Six found in most of the literature related to emotional speech [50]. EMOVO is a perfectly balanced dataset made up of 588 audio samples.

In Figure 2 we show the duration of the conversations per sentiment. As we can see, they all appear to be quite balanced and of the same length. The sentences were designed with the emotionally neutral semantic content so as not to generate bias in the recognition of emotions by both machines and humans. EMOVO has all the phonemes of the Italian language, and all its sentences are characterised by a fair balance between voiced and unvoiced consonants.

The performances of the actors were recorded in the laboratories of the Fondazione Ugo Bordoni in Rome with professional equipment by using a sampling frequency of 48 kHz. The recordings were saved in the *WAV* format.

## 4 | METHODS

The workflow of our implemented methodology can be summarised into four main steps: data collection, data transformation, modelling and testing. The workflow is depicted in Figure 3. All of the code of our project is available at <https://github.com/fp1acm8/SER>.

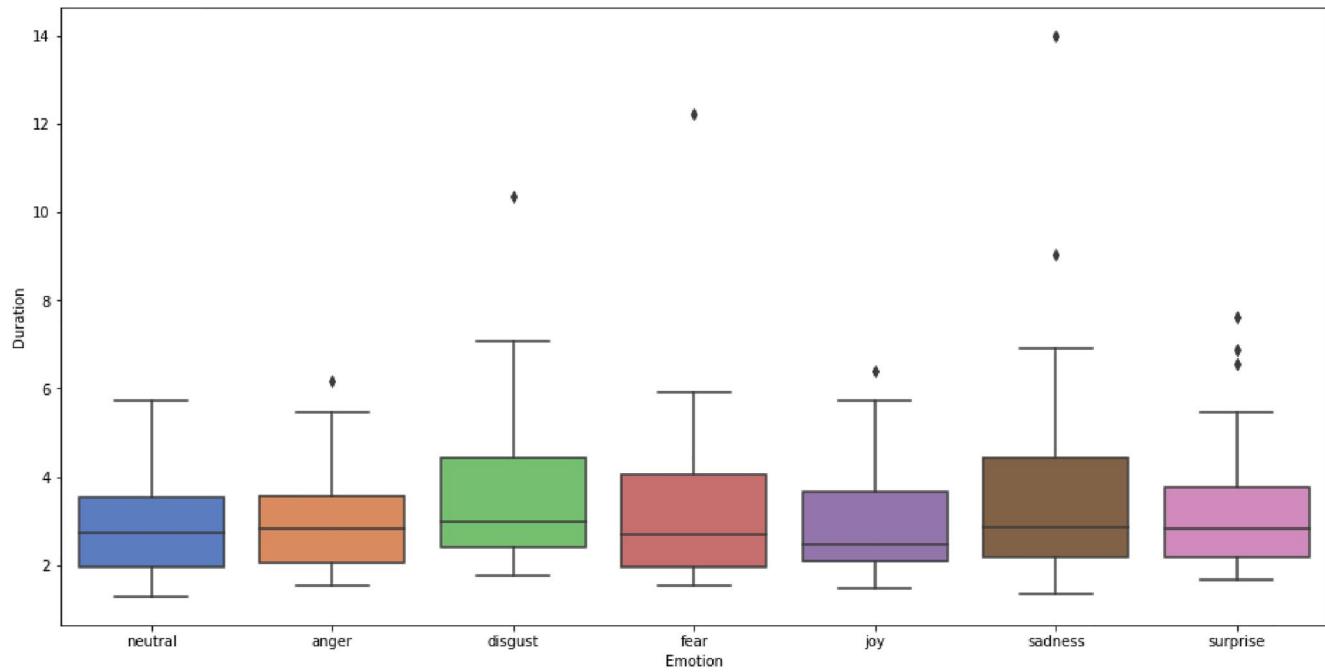
In the following subsections we will describe each of the steps performed.

### 4.1 | STEP 1: Data collection

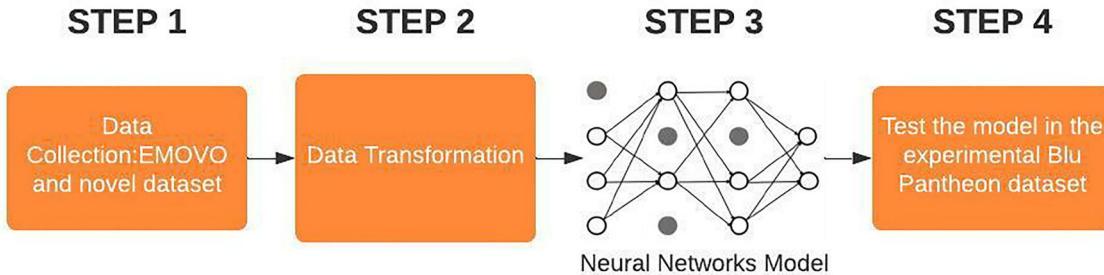
In this first phase we performed a thorough search to identify a possible Italian labelled dataset to be used in our study for training. As the novel dataset provided was unlabelled, so we decided to look for labelled dataset on which we could train a machine learning model to be later tested in the novel context.

<sup>1</sup>Gartner is a global research and advisory firm providing information, advice, and tools for leaders in IT, finance, HR, customer service and support, communications, legal and compliance, marketing, sales, and supply chain functions.

<sup>2</sup>Waveform Audio File Format (WAVE or WAV due to its filename extension).



**FIGURE 2** Boxplot showing the duration of the conversations per emotion. In this way, we can see how the dataset is well balanced among the examples of different emotions.



**FIGURE 3** Workflow of our experimental setting.

Language was not the only feature to consider when we looked for publicly available dataset. In fact, we also tried to look for dataset where there was a correspondence between the emotions that we wanted to identify in our model, considering the novel dataset and the contact tracing application.

For example, in the context in which we developed our project emotions such as ‘happy’ or ‘joy’ are not something we should expect in a conversation between COVID-19 positive patients and a virtual agent.

Moreover, we looked for dataset where there was a similar duration of the audio samples collected to have a correspondence between distributions in the training and test set. This is because the duration of an audio sample directly affects the amount of information extracted from the audio signal features. Therefore, having datasets with audio samples of extremely different duration could have affected the performance of our model.

Last, but not least, it is important to take into account the quantity and the quality of the collected data. It is crucial, in

fact, to make sure that the audio samples collected are high quality data, not affected by excessive noise. For this reason, among the possible publicly available solutions, we decided to use the EMOVO dataset (described in Section 3) to train the SER model.

## 4.2 | STEP 2: Data transformation

We performed data transformation steps in order to increase the similarity between EMOVO and our experimental dataset.

Firstly, we decided to remove the emotion joy from the EMOVO dataset (as this was an emotion not pertinent to our experimental dataset). We further combined similar emotional states in a unique class (anger and disgust were merged in a unique class named disappointed).

Moreover, we performed data augmentation by using techniques such as noise addition and change the pitch of the conversation.

Data were standardised using the  $z - score$  normalisation and an extensive set of data cleaning steps were performed in the novel dataset. In particular, the following cleaning and data filtering steps were performed on our novel dataset. First, we deleted audio samples with a duration longer than 14 s that is, those audio with sampling errors made by the system developed by Blu Pantheon. Secondly, we deleted audio samples for which the question asked by the bot could not be retrieved. Such cleaning and transformation steps led us to obtain a dataset made of 2871 observations, which represented 96% of the samples originally provided to us by Blu Pantheon. These cleaning and transformation steps led us to obtain a dataset made of 2871 observations, which represented 96% of the samples originally provided to us by Blu Pantheon. Moreover, in our novel dataset, we performed extensive processing steps, merging the BOT and the user answer, as well as classifying the questions made by the BOT into seven classes: clinical interview, tracing, telephone number, name of the contact, social interview, guidelines and repeat.

### 4.3 | STEP 3: Modelling

In this central steps, we performed mainly two tasks: features extraction and machine learning modelling of the features. We will describe the two steps separately describing the methodologies used.

#### 4.3.1 | Features extraction

First of all, we extracted the relevant features to be used as input to our classification model. In particular, we decided to use the python library librosa to extract the following features from the audio data:

- Chroma vector: a 12 element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music
- Root mean square (RMS) value for each frame from the audio samples
- Spectral features: Spectral flatness, spectral centroid, spectral bandwidth and spectral rolloff
- Zero-crossing rate
- The Mel-frequency cepstral coefficient

The final dimension of the EMOVO dataset was 1754 observations and 39 features. The choice of such features, rather than others which are commonly used in speech experiments (such as the Mel Frequency Spectrograms) was mainly driven by the dataset dimension. Having only a small amount of labelled data (the EMOVO dataset) did not allow to properly train an ML model with such features, and this is the reason why instead of spectrogram information we decided to use the set of features described above.

### 4.4 | Machine learning modelling

In our project, the chosen ML architecture is consisted in a Multilayer Perceptron (MLP), which we will describe more in details in Section 4.5 Moreover, we compared the performances of the MLP to two further extremely famous ML methods: the Support Vector Classification and The Extreme Gradient Boosting (XGBR) method. Support vector machines (SVMs) represent among the most powerful machine learning models, developed since 1995 and they are based on the VC theory (so called as proposed by Vapnik and Chervonenkis [51]). They can be used successfully both for regression and for classification. Since in our case, we used them for classification. We used the acronym SVC: Support Vector Classification. Also XGBR represents one of the most popular algorithms for classification and regression nowadays (since the introduction in 2015) and is based on the gradient boosting algorithm [52]. For more details, please check the reference paper [52].

### 4.5 | Multilayer perceptron

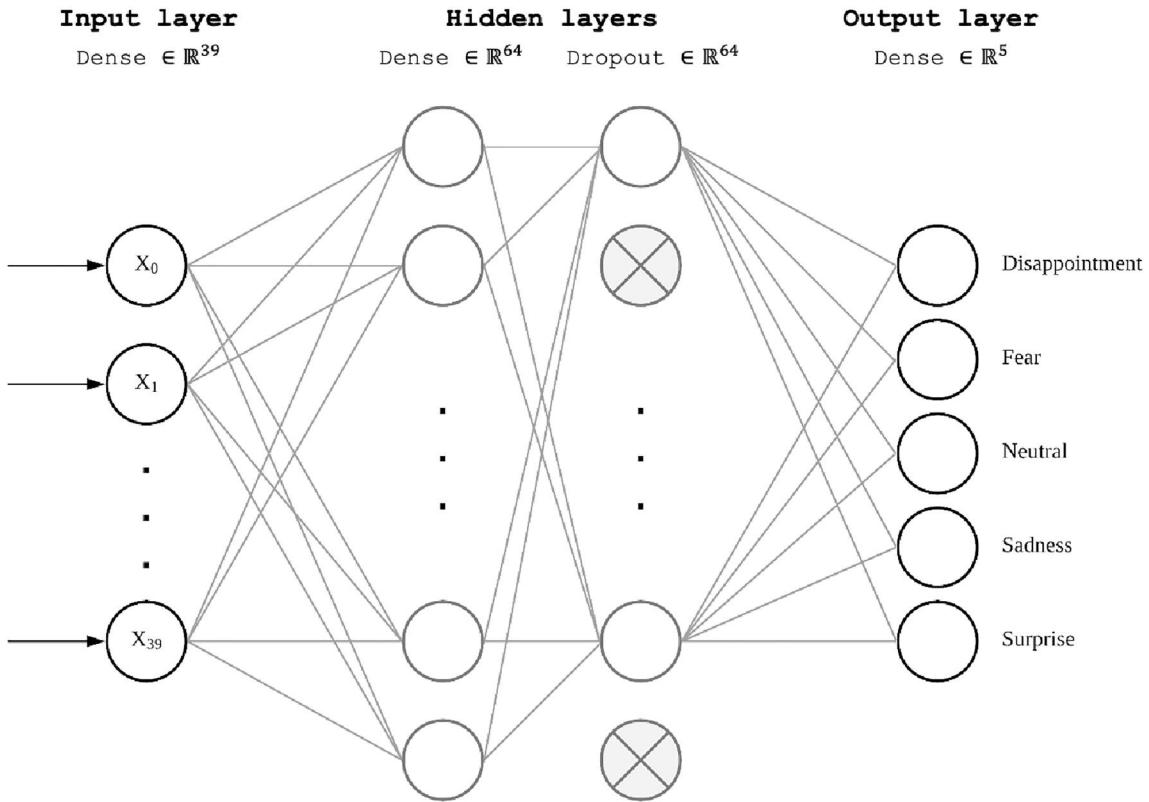
Since their introduction in the 80s, neural networks models have proved to be extremely successful in performing a wide variety of different classification and regression tasks [24] and have been successfully applied to several different fields from biology to natural language processing, from object detection to scene classification [53–56]. In our project, we implemented a classifier based on a multilayer perceptron (MLP) neural network. The architecture implemented is made of 39 units in the input layer with Relu activation function, a dense and a dropout hidden layer and a final dense output layer. We report the implemented neural network scheme in Figure 4. We used Adam optimiser and categorical cross-entropy loss function. The categorical cross-entropy loss function is defined as follows:

$$\text{Loss} = - \sum_{i=1}^n y_i \cdot \log \hat{y}_i \quad (1)$$

where  $n$  is the output size (i.e., the number of label classes),  $y_i$  is the target value and  $\hat{y}_i$  is the  $i$ -th scalar value in the model output. In particular  $y_i$  and  $\hat{y}_i$  are probabilities associated respectively to the true and predicted  $i$ -th class. We used 10-folds-cross validation, batch size of 32, 10 epochs and early stopping. To implement the MLP architecture, we used the Keras python library.

### 4.6 | STEP 4: Test the model in the experimental dataset

In step 4, we tested the trained model on EMOVO in our new unlabelled experimental dataset. An emotion was predicted for each new sample.



**FIGURE 4** Representation of the multilayer perceptron architecture (MLP) with dropout represented by the nodes with a cross in the second hidden layer.

Moreover, we proceeded in the following way. Since we had also time and the user id information, we merged the various pieces of conversation of a single user, considering to obtain the overall stream of a conversation. Knowing that an entire conversation is made up of multiple audio samples, we actually predict the emotional state of a single user several times based on the length of the conversation (i.e. the number of audio samples per user). We therefore made the assumption that it is unlikely that a user during a conversation of a few minutes will change his emotional state many times. So if the model predicts 3/4 different emotions for a single user, we could conclude that the prediction will be inconsistent. While this assumption is plausible, it does not sufficiently explain the results obtained. Therefore we explored the data further through data visualisation techniques (see Results and Experiments section). For instance, investigating if there were any patterns in the predictions depending on the sentence pronounced by the user or by the agent.

## 5 | EXPERIMENTS AND RESULTS

### 5.1 | Training on EMOVO dataset

We first trained our model on the EMOVO dataset. For evaluating performances in the EMOVO dataset, we used a 10-fold cross-validation approach, with 90% of this data was used to train the dataset, while the remaining 10% was used as a test.

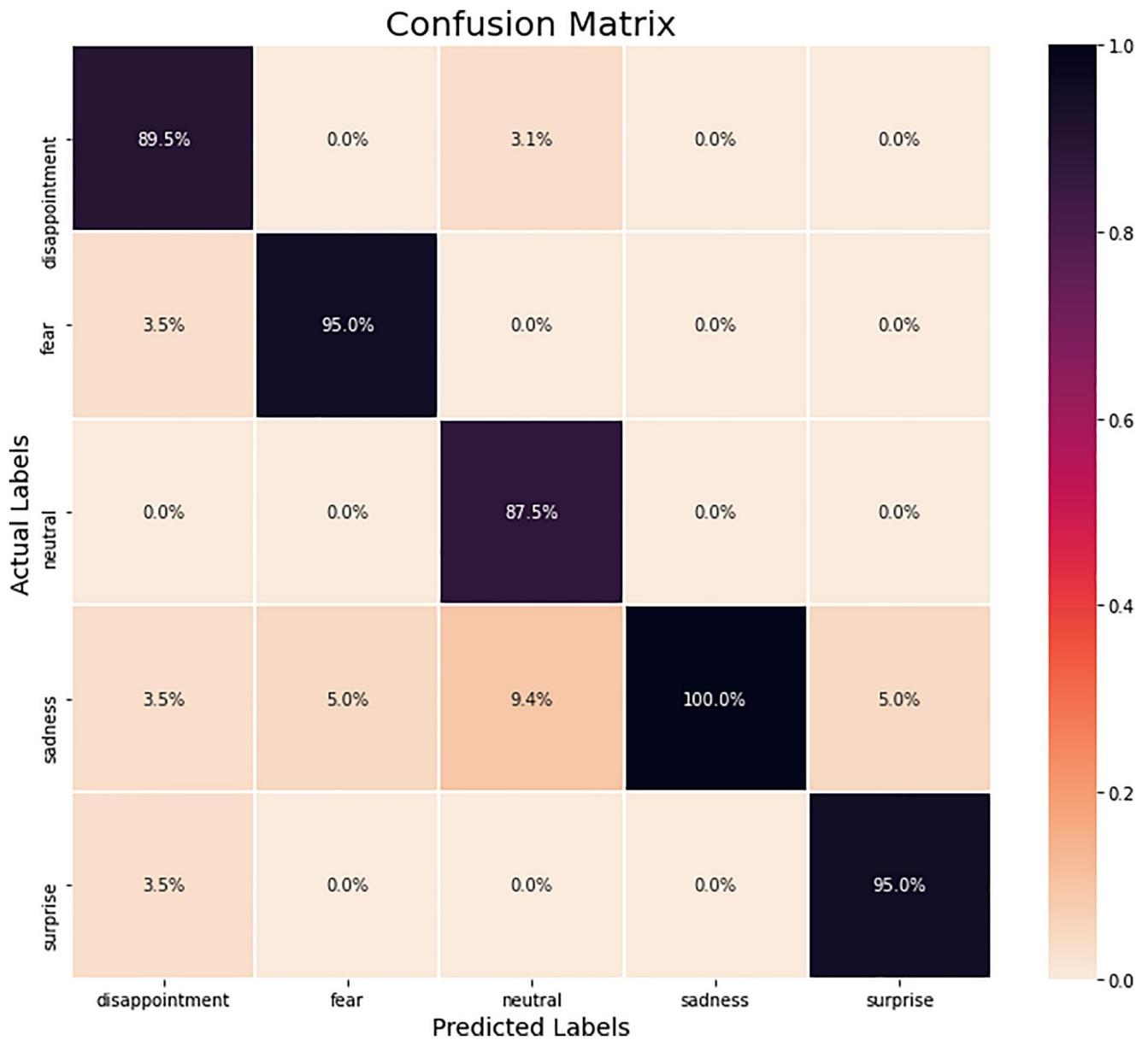
**TABLE 1** In the table, we show the performances in terms of precision, recall, F-Score and accuracy for the test folder in the 10-fold cross validation.

Metric	XGB	SVC	MLP
Precision	$0.81 \pm 0.11$	$0.80 \pm 0.06$	<b><math>0.93 \pm 0.05</math></b>
Recall	$0.74 \pm 0.13$	$0.82 \pm 0.09$	<b><math>0.91 \pm 0.09</math></b>
FScore	$0.76 \pm 0.08$	$0.84 \pm 0.03$	<b><math>0.92 \pm 0.03</math></b>
Accuracy	$0.76 \pm 0.13$	$0.80 \pm 0.09$	<b><math>0.92 \pm 0.05</math></b>

*Note:* We report the mean and standard deviation over the 10 test folds. Bold values represent the best performing cases.

The training results obtained were promising, in terms of accuracy obtaining a mean accuracy of 0.92 (0.02 standard deviation). In Table 1 we report the test set performances (mean and standard deviation in the 10 fold). We report them for the MLP, the SVC and the XGB classifier implemented as baseline experiments (and which we described in the methods section). Performances were evaluated according to the following performance indicators: accuracy, precision, recall and F1 score (we report their definition in the Supplementary Material S1).

As we can see from Table 1, the MLP outperformed in all of the performance metrics than the other two machine learning models implemented. Therefore we chose the MLP as the optimal model and proceeded in testing it in our novel dataset from Blu Pantheon. In Figure 5 we also present the confusion matrix with the mean accuracies over the 10 folds obtained in the Emovo test set.



**FIGURE 5** Average accuracies of the test set in the 10-folds cross validation, divided per emotion.

## 5.2 | Testing on Blu Pantheon novel dataset

We therefore proceeded using the fitted model to test it in our novel dataset. This was possible due to the high similarity between the EMOVO and the original dataset, in terms of audio samples lengths and distributions.

Comparing the two datasets it can be noticed that the two datasets distributions are very similar up to the 75% percentile, in terms of sounds tracks. The minimum duration, the median and the 25% percentile, differs only by a few hundredths of a second. A difference of 1.10 s between EMOVO and the experimental dataset can be found in the mean value of the duration of the respective audio samples. This difference is due to the fact that in the experimental dataset, there are audio samples with a duration of up to 57 s. However, inspecting these samples, it was noticed that the long durations are often

due to sampling errors made by the system developed by Blu Pantheon.

For instance, in the 57.64 s recording, the user answers the question asked by the virtual agent in the first few seconds of the recording while the system continues to record background noises thinking it is the user who is answering.

Following these considerations, we decided to discard all those defective records assuming the same maximum duration as EMOVO (i.e. 14 s). In this way, the difference between the mean values drops to 0.72 s, explained by the higher standard deviation in the case of the experimental dataset.

In conclusion, we can say that from the point of view of the duration, the two datasets are similar enough to justify the use of the model trained on EMOVO to be tested later on the experimental dataset by using a transfer learning approach.

### 5.3 | Experimental sentiment analysis of the novel Blu Pantheon dataset

The first experimental analysis we performed was per user (Subsection 5.3.1). Subsequently, we analysed the predictions more in depth on the basis of the question asked by the virtual agent (Subsection 5.3.2).

Eventually, we made some further qualitative evaluation, verifying whether by listening to some audio samples it was possible to distinguish the different predicted emotions (Subsection 5.3.3).

#### 5.3.1 | Analysis by user

In this section we analysed prediction grouping by users.

Each user corresponds to a certain conversation, and each conversation is made up of multiple audio samples. So it can happen that for a single user, several predictions of his emotional state have been made (remember that the model predicts an emotion for each audio sample provided).

We therefore decided to associate each user with the *mode* among all the emotional states predicted for him/her. For example, suppose the model predicts five times ‘neutral,’ three times ‘fear’ and two times ‘disappointment’ for a generic user  $x$ .

Then the emotional state associated with the user  $x$  is ‘neutral.’ In this way, we have only one prediction per user for a total of 353.

A summary representation can be found in Figure 6. The prediction made is the mode of all the predicted emotions for a single user, so the emotion associated with a certain user is not

necessarily the same throughout the conversation but it can change.

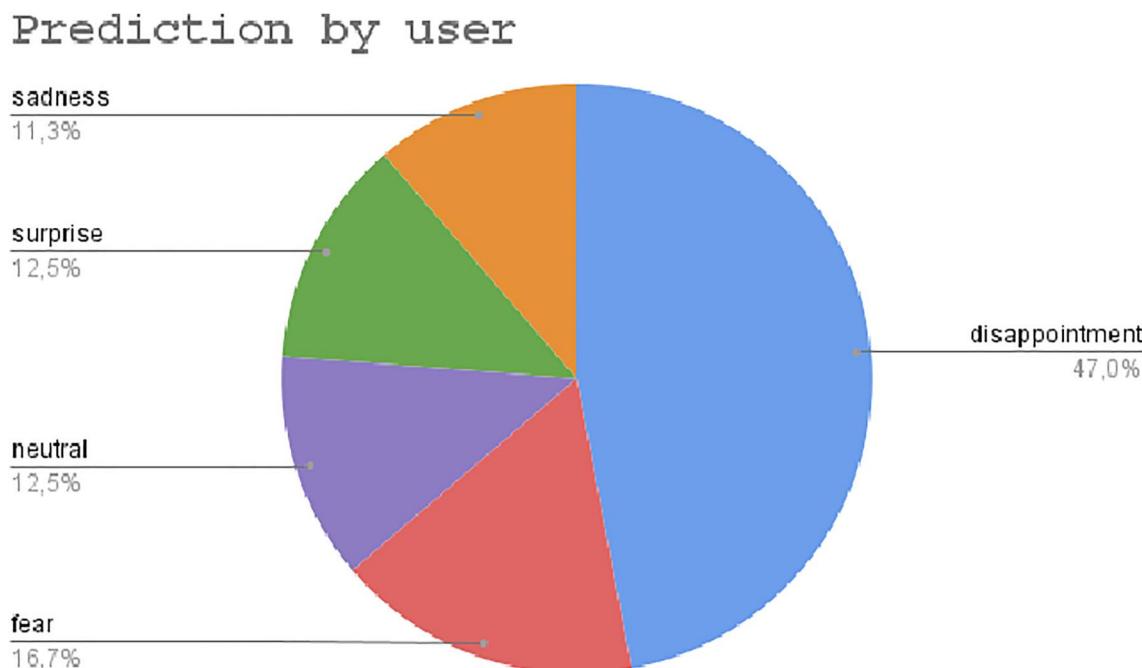
Only in 14% of the cases observed (i.e. 48 cases out of 353), the emotion remained unchanged. A summary representation of these cases, for which the model predicted only one emotion during the whole conversation, can be seen in Figure 7. From these two graphs, we see how ‘disappointment’ and ‘sadness’ are the emotions most likely to be predicted for the entire duration of the conversation. This concept of uniqueness of prediction is important because, as we mentioned before, it is unlikely that a user will experience three or four different emotions during a 10 min conversation. On the other hand, if only one emotion is detected during the entire conversation, then it will be likely that the prediction made for that user is reliable.

At the same time, this does not mean that the model does not work properly. There may be nuances in the user’s voice depending on the question asked by the virtual agent that cause one emotion to be predicted rather than another. It is therefore important to analyse the emotions also on the basis of the questions received by the users.

#### 5.3.2 | Analysis by sentence

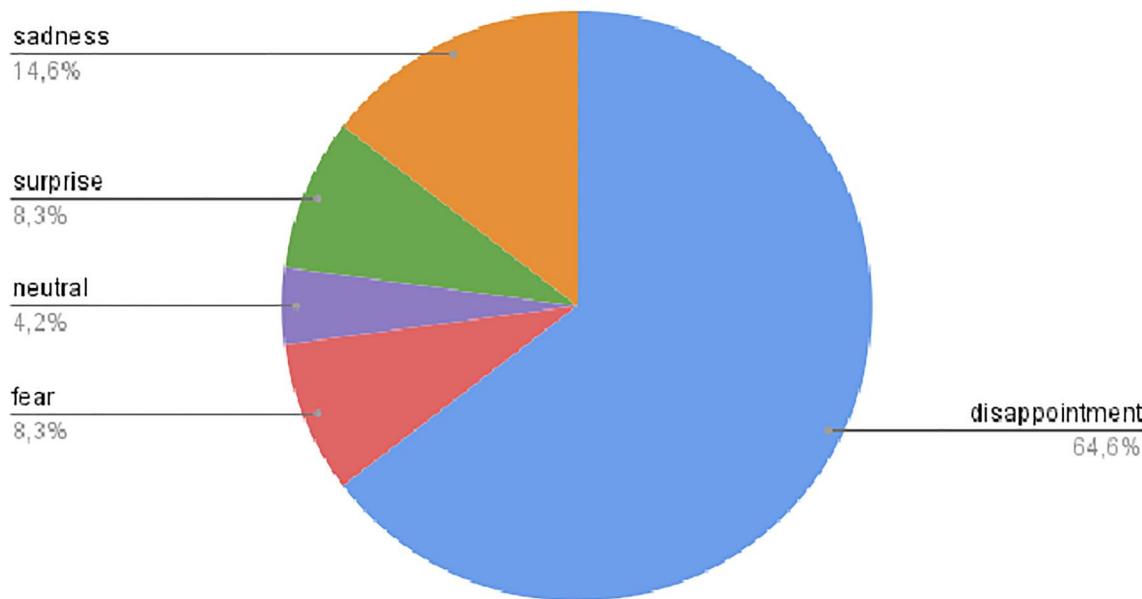
Sentiment analysis by sentence was carried out by analysing the predicted emotions according to the questions asked by the virtual agent which we have divided into seven classes defined in Section 3.

Six of these classes (i.e. Clinical interview, Tracing, Telephone number, Name of the contact, Social interview and



**FIGURE 6** Pie chart of emotions by the user obtained through the mode of all predictions made for that user.

## Prediction by user (unique emotion)



**FIGURE 7** Pie chart of emotions per user for which a single emotion was predicted throughout the conversation.

Guidelines) represent a specific section of the conversation between the patient and virtual agent. With this classification, it is possible to keep track at a macro level of the evolution of emotions.

The six classes have in fact a precise temporal order within the conversation (e.g. first a clinical interview is carried out with the patient, then the close contacts are traced up to the social interview and the remainder of the instructions to follow during the quarantine).

The 7th class, on the other hand, is ‘Repeat.’ Sometimes, in fact, it happens that the bot is unable to correctly isolate the answer given by the user and therefore rephrases the question. We therefore decided to enter this category to see if repeating the question significantly alters the patient’s emotional state, highlighting a clear disservice to be solved by the company.

We present the results in Figure 8. In the stacked bar chart, each column is a question class. Each bar is also divided into five distinct parts and the five different colours that represent the percentage of predicted emotions for each class. The columns are arranged in chronological order with the class ‘Repeat’ left for last. The analysis shows how the emotional state of disappointment prevails over the others; however, this should not be interpreted exclusively as dissatisfaction with the service, but more generally as a more decisive tone of voice.

In fact, it is typical to respond to a virtual agent articulating the words well in order to be sure that they have been understood.

This can be found above in the ‘Telephone number’ and ‘Name of the contact’ columns which are the key questions of the service and in which the degree of disappointment rises while sadness decreases. Similar to the considerations made for

‘disappointment,’ ‘sadness’ identifies a lower tone of voice that could also indicate a certain level of boredom and little involvement in the conversation.

Moreover, it is interesting to see the behaviour of ‘surprise’ and ‘fear’ that rise in percentage terms, especially towards the end of the conversation.

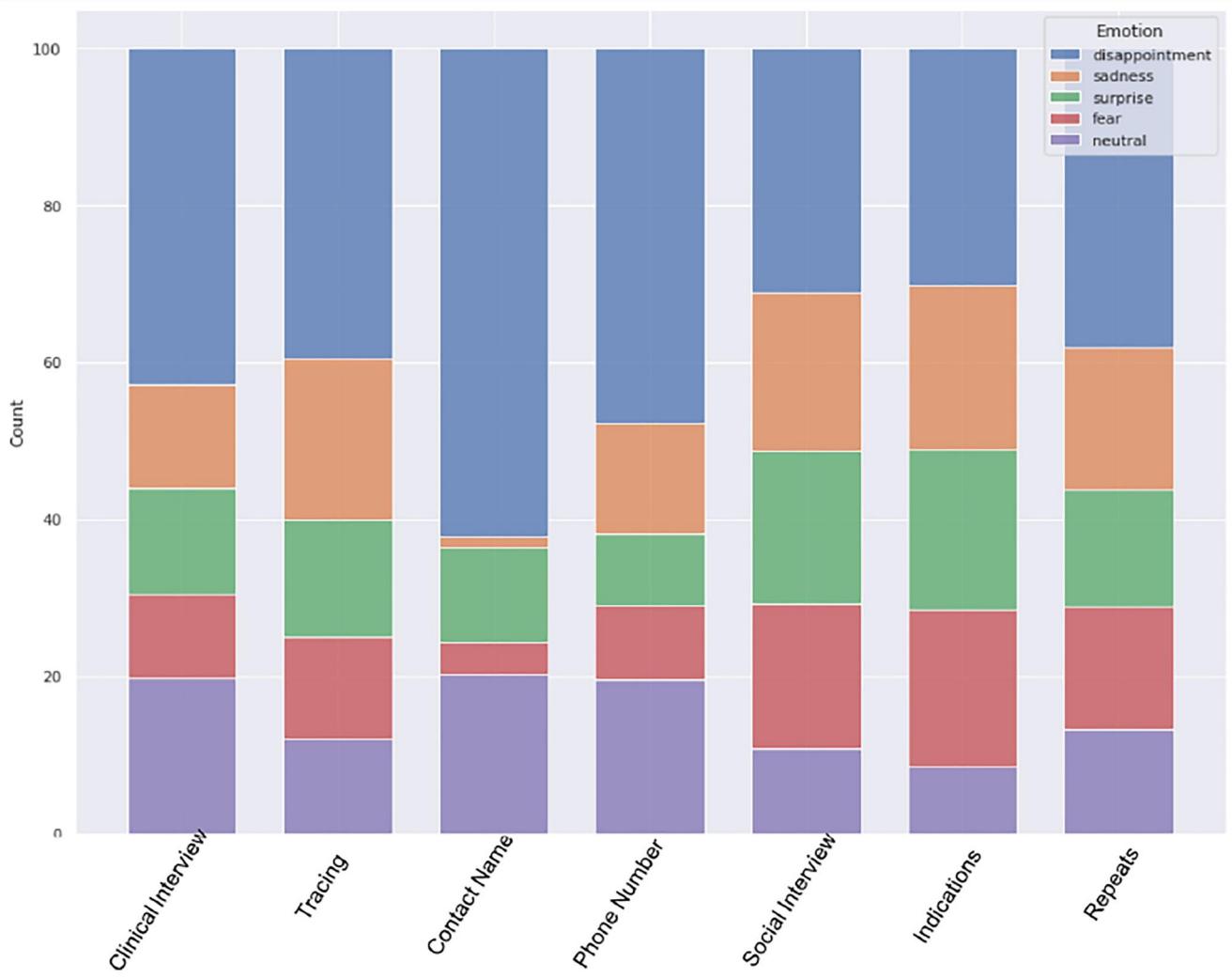
In particular, when the bot asks the user if there have been other contacts outside the family environment and if he wants to listen to the instructions to follow during the quarantine again.

While in the first case, it is immediate to understand the reason for the surprise or fear (i.e. either because the user does not have an immediate answer or he is afraid of putting other people in unpleasant situations), in the case of the quarantine indications, the result can be explained by the fact that the bot after a long series of indications in which the patient could get distracted, he is asked if he wants to listen to the indications again, finding him sometimes displaced. Finally, as regards the ‘Repeat’ class, it is noted that the results are in line with the average values achieved for the other classes.

This means that there is insufficient evidence to show that repeating a response adversely alters the patient’s emotional state.

### 5.3.3 | Empirical analysis

By empirical analysis, we mean an analysis carried out by directly listening to some of the audio samples and indicating whether the emotion predicted by the model is actually detectable by the human ear. Specifically, we sampled 15 random audio samples for each emotional state for a total of 75 plays.



**FIGURE 8** Stacked bar chart coloured by emotion with questions on the *x*-axis and the percentage count on the *y*-axis.

Listening to the audio samples, due to the short and pre-defined answers, it was not possible to clearly distinguish between emotions.

However, one could hear tonal differences between the various classes as hypothesised previously (Subsection 5.3.2). For example, ‘disappointment’ generally appeared to have a more decisive tone than sadness.

Therefore it can be concluded from the analysis made that the model shows some promising results about its correct functioning but some structural problems such as the guided and brief answers that the user is forced to give to the bot do not allow to fully evaluate the performance of the model. In this regard, it is recommended to create sections of the call where the user can express himself more freely.

This can help identify the patient's emotional state more accurately.

Furthermore, the analysis could be improved by sending users a questionnaire aimed at tracking their emotional state. If we cross-reference the results of the questionnaire and the predictions made by the model, we could achieve better results and more specific insights.

## 6 | CONCLUSIONS

In this work, we present a neural network-based model to predict emotions in a chatbot developed for contact tracing during COVID-19. The importance of the work presented in this manuscript is manifold.

In particular, considering the emotions recognised by the system, the following are taken.

- Provide further assistance to patients with ‘at risk’ emotional states and in need of psychological support. For example, if fear or sadness is detected, it can be decided to entrust the case to a human operator who is more specialised in the treatment of specific patients.
- Adapt the language and the type of questions asked by the virtual agent based on the patient’s emotional state. At present, the flow of the conversation and the questions asked by the virtual agent follow a tree pattern dependent on the patient’s responses. This pattern can be redesigned based on the emotions identified during the conversation.

- More specific questions can be created to assess the patient's emotional state.
- Highlight possible disservices during the call and remedy those. A quality service would have an immediate positive economic impact for the supplier company and its customers and also positive effects on the collective trust in Human-Computer Interaction (HCI) applications.

In conclusion, the sentiment analysis carried out with the methodology proposed by this thesis would greatly help improve the quality of the service which means not only an immediate positive economic impact for the supplier company Blu Pantheon and its customers but also positive effects on the collective trust in Human-Computer Interaction (HCI) applications.

HCI solutions oriented towards sentiment analysis assumed increasing importance during the COVID-19 pandemic crisis, where authorities mandated both public and private organisations to embrace new practices for working remotely and maintaining social distancing.

The model implemented showed very promising results. An in-depth analysis of patients' emotional states during the call can, in fact, help the company significantly improve the service and provide an extremely important decision support system tool.

Future work may include the collection of new audio samples, and therefore, possible implementation of deep learning and multi-modal approaches for the identification of emotions in voice-bot conversation.

## CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

## DATA AVAILABILITY STATEMENT

Code available at: <https://github.com/fp1acm8/SER> Data available upon requests to the authors.

## ORCID

Giovanna Maria Dimitri  <https://orcid.org/0000-0002-2728-4272>

## REFERENCES

1. Covid-19 OED Online (2021, September). Oxford University Press. <https://www.oed.com/viewdictionaryentry/Entry/88575495> (2021). Accessed September 2021
2. Page, J., Hinshaw, D., McKay, B.: Hunt for Covid-19 Origin, Patient Zero Points to Second Wuhan Market—The man with the first confirmed infection of the new coronavirus told the WHO team that his parents had shopped there. Wall St. J. (2021). <https://www.wsj.com/articles/in-hunt-for-covid-19-origin-patient-zero-points-to-second-wuhan-market-11614335404>
3. Ministero della Salute FAQ - Covid-19 domande e risposte. <https://www.salute.gov.it> (2021)
4. blupantheon.com (2022)
5. Bianchini, M., et al.: Deep neural networks for structured data. In: Computational Intelligence for Pattern Recognition, pp. 29–51. Springer (2018)
6. Bishop, C.M., Nasrabadi, N.M.: Pattern Recognition and Machine Learning, vol. 4. Springer (2006)
7. Flach, P.: Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge university press (2012)
8. Dimitri, G.M., et al.: Unsupervised stratification in neuroimaging through deep latent embeddings. In: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 1568–1571. IEEE (2020)
9. Miconi, F., Dimitri, G.M.: A Machine Learning Approach to Analyse and Predict the Electric Cars Scenario: The Italian Case. Plos one (2023)
10. Zhang, J., et al.: Emotion recognition using multi-modal data and machine learning techniques: a tutorial and review. Inf. Fusion 59, 103–126 (2020). <https://doi.org/10.1016/j.inffus.2020.01.011>
11. Rekanar, K., et al.: Sentiment analysis of user feedback on the HSE's Covid-19 contact tracing app. Ir. J. Med. Sci. 1971(1-10), 103–112 (2022). <https://doi.org/10.1007/s11845-021-02529-y>
12. Ouellet, S.: Real-time emotion recognition for gaming using deep convolutional network features. arXiv preprint arXiv:14083750 (2014)
13. Oh, K.J., et al.: A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation. In: 2017 18th IEEE International Conference on Mobile Data Management (MDM), pp. 371–375. IEEE (2017)
14. Kalantarian, H., et al.: Labeling images with facial emotion and the potential for pediatric healthcare. Artif. Intell. Med. 98, 77–86 (2019). <https://doi.org/10.1016/j.artmed.2019.06.004>
15. Braun, M., Weber, F., Alt, F.: Affective automotive user interfaces—Reviewing the state of emotion regulation in the car. arXiv preprint arXiv:200313731 (2020)
16. Petrushin, V.: Emotion in speech: recognition and application to call centers. In: Proceedings of Artificial Neural Networks in Engineering, vol. 710, pp. 22 (1999)
17. Ziser, Y., Kravi, E., Carmel, D.: Humor detection in product question answering systems. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 519–528 (2020)
18. Anagnostopoulos, C.N., Iliou, T., Giannoukos, I.: Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011. Artif. Intell. Rev. 43(2), 155–177 (2015). <https://doi.org/10.1007/s10462-012-9368-5>
19. Cao, H., et al.: Crema-d: Crowd-sourced emotional multimodal actors dataset. IEEE Trans. Affect. Comput. 5(4), 377–390 (2014). <https://doi.org/10.1109/taffc.2014.2336244>
20. Livingstone, S.R., Russo, F.A.: The Ryerson audio-visual database of emotional speech and Song (RAVDESS): a dynamic, multimodal set of facial and vocal expressions in North American English. PLoS One 13(5), e0196391 (2018). <https://doi.org/10.1371/journal.pone.0196391>
21. Jackson, P., Haq, S.: Surrey Audio-Visual Expressed Emotion (Savee) Database. University of Surrey Guildford, UK (2014)
22. Dupuis, K., Pichora-Fuller, M.K.: Toronto Emotional Speech Set (Tess)-younger Talker\_happy (2010)
23. Busso, C., et al.: IEMOCAP: interactive emotional dyadic motion capture database. Comput. Humanit. 42(4), 335–359 (2008). <https://doi.org/10.1007/s10579-008-9076-6>
24. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015). <https://doi.org/10.1038/nature14539>
25. Larranaga, P., et al.: Machine learning in bioinformatics. Briefings Bioinf. 7(1), 86–112 (2006). <https://doi.org/10.1093/bib/bbk007>
26. Spiga, O., et al.: Machine learning application for patient stratification and phenotype/genotype investigation in a rare disease. Briefings Bioinf. 22(5), bbaa434 (2021). <https://doi.org/10.1093/bib/bbaa434>
27. Sebe, N., et al.: Machine Learning in Computer Vision, vol. 29. Springer Science & Business Media (2005)
28. Vea, C., et al.: Interactive alkaptonuria database: investigating clinical data to improve patient care in a rare disease. Faseb. J. 33(11), 12696–12703 (2019). <https://doi.org/10.1096/fj.201901529r>
29. Dimitri, G.M., et al.: Modeling brain–heart crosstalk information in patients with traumatic brain injury. Neurocritical Care 36(3), 738–750 (2022). <https://doi.org/10.1007/s12028-021-01353-7>
30. Soleymani, M., et al.: A survey of multimodal sentiment analysis. Image Vis Comput. 65, 3–14 (2017). <https://doi.org/10.1016/j.imavis.2017.08.003>

31. Selvaraj, M., Bhuvana, R., Karthik, S.P.: Human speech emotion recognition. *Int. J. Eng. Technol.* 8, 311–323 (2016)
32. Shaikh Nilofer, R., et al.: Automatic emotion recognition from speech signals: a Review. *Int. J. Sci. Eng. Res.* 6(4) (2015)
33. Lieskovská, E., et al.: A review on speech emotion recognition using deep learning and attention mechanism. *Electronics* 10(10), 1163 (2021). <https://doi.org/10.3390/electronics10101163>
34. Nwe, T.L., Foo, S.W., De Silva, L.C.: Speech emotion recognition using hidden Markov models. *Speech Commun.* 41(4), 603–623 (2003). [https://doi.org/10.1016/s0167-6393\(03\)00099-2](https://doi.org/10.1016/s0167-6393(03)00099-2)
35. Shen, P., Changjun, Z., Chen, X.: Automatic speech emotion recognition using support vector machine. In: Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, vol. 2, pp. 621–625. IEEE (2011)
36. Shaw, A., Vardhan, R.K., Saxena, S.: Emotion recognition and classification in speech using artificial neural networks. *Int. J. Comput. Appl.* 145(8), 5–9 (2016). <https://doi.org/10.5120/ijca2016910710>
37. Huang, Z., et al.: Speech emotion recognition using CNN. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 801–804 (2014)
38. Kwon, S., et al.: MLT-DNet: speech emotion recognition using 1D dilated CNN based on multi-learning trick approach. *Expert Syst. Appl.* 167, 114177 (2021). <https://doi.org/10.1016/j.eswa.2020.114177>
39. Baevski, A., et al.: wav2vec 2.0: a framework for self-supervised learning of speech representations. *Adv. Neural Inf. Process. Syst.* 33, 12449–12460 (2020)
40. Hsu, W.N., et al.: Hubert: self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio, Speech, and Lang. Proces.* 29, 3451–3460 (2021). <https://doi.org/10.1109/taslp.2021.3122291>
41. Castellano, G., De Carolis, B., Macchiarulo, N.: Automatic facial emotion recognition at the COVID-19 pandemic time. *Multimed. Tool. Appl.*, 1–19 (2022). <https://doi.org/10.1007/s11042-022-14050-0>
42. Yang, B., Jianming, W., Hattori, G.: Face mask aware robust facial expression recognition during the COVID-19 pandemic. In: 2021 IEEE International Conference on Image Processing (ICIP), pp. 240–244. IEEE (2021)
43. Scarpina, F.: Detection and recognition of fearful facial expressions during the coronavirus disease (COVID-19) pandemic in an Italian sample: an online experiment. *Front. Psychol.* 11, 2252 (2020). <https://doi.org/10.3389/fpsyg.2020.02252>
44. Huang, Z., et al.: Public perception of the use of digital contact-tracing tools after the COVID-19 lockdown: sentiment analysis and opinion mining. *JMIR Format. Res.* 6(3), e33314 (2022). <https://doi.org/10.2196/33314>
45. Srikanth, J., et al.: Sentiment analysis on COVID-19 Twitter data streams using deep Belief neural networks. *Comput. Intell. Neurosci.* 2022, 2022–11 (2022). <https://doi.org/10.1155/2022/8898100>
46. Alhuzali, H., et al.: Emotions and topics expressed on Twitter during the COVID-19 pandemic in the United Kingdom: comparative geolocation and text mining analysis. *J. Med. Internet Res.* 24(10), e40323 (2022). <https://doi.org/10.2196/40323>
47. Choudrie, J., et al.: Applying and understanding an advanced, novel deep learning approach: a Covid 19, text based, emotions analysis study. *Inf. Syst. Front.* 23(6), 1431–1465 (2021). <https://doi.org/10.1007/s10796-021-10152-6>
48. [www.braincontrol.eu](http://www.braincontrol.eu) (2022)
49. Costantini, G., et al.: EMOVO corpus: an Italian emotional speech database. In: International Conference on Language Resources and Evaluation (LREC 2014), pp. 3501–3504. European Language Resources Association (ELRA) (2014)
50. Cowie, R., Cornelius, R.R.: Describing the emotional states that are expressed in speech. *Speech Commun.* 40(1–2), 5–32 (2003). [https://doi.org/10.1016/s0167-6393\(02\)00071-7](https://doi.org/10.1016/s0167-6393(02)00071-7)
51. Chapelle, O., et al.: Choosing multiple parameters for support vector machines. *Mach. Learn.* 46(1), 131–159 (2002). <https://doi.org/10.1023/a:1012450327387>
52. Chen, T., et al.: Xgboost: extreme gradient boosting. R Package Version 04-2 1(4), 1–4 (2015)
53. Dimitri, G.M., et al.: Multimodal and multicontrast image fusion via deep generative models. *Inf. Fusion* 88, 146–160 (2022). <https://doi.org/10.1016/j.inffus.2022.07.017>
54. Otter, D.W., Medina, J.R., Kalita, J.K.: A survey of the usages of deep learning for natural language processing. *IEEE Transact. Neural Networks Learn. Syst.* 32(2), 604–624 (2020). <https://doi.org/10.1109/tnnls.2020.2979670>
55. Goldberg, Y.: A primer on neural network models for natural language processing. *J. Artif. Intell. Res.* 57, 345–420 (2016). <https://doi.org/10.1613/jair.4992>
56. Nogueira, K., Penatti, O.A., Dos Santos, J.A.: Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recogn.* 61, 539–556 (2017). <https://doi.org/10.1016/j.patcog.2016.07.001>

## SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

**How to cite this article:** Pucci, F., Fedele, P., Dimitri, G.M.: Speech emotion recognition with artificial intelligence for contact tracing in the COVID-19 pandemic. *Cogn. Comput. Syst.* 1–15 (2023). <https://doi.org/10.1049/ccs2.12076>

Article

# Text-Based Emotion Recognition in English and Polish for Therapeutic Chatbot

Artur Zygadło \*, Marek Kozłowski  and Artur Janicki 

Faculty of Electronics and Information Technology, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland; Marek.Kozlowski@pw.edu.pl (M.K.); Artur.Janicki@pw.edu.pl (A.J.)

\* Correspondence: zyga@artur@google.com

**Abstract:** In this article, we present the results of our experiments on sentiment and emotion recognition for English and Polish texts, aiming to work in the context of a therapeutic chatbot. We created a dedicated dataset by adding samples of neutral texts to an existing English-language emotion-labeled corpus. Next, using neural machine translation, we developed a Polish version of the English database. A bilingual, parallel corpus created in this way, named CORTEX (CORpus of Translated Emotional teXts), labeled with three sentiment polarity classes and nine emotion classes, was used for experiments on classification. We employed various classifiers: Naïve Bayes, Support Vector Machines, fastText, and BERT. The results obtained were satisfactory: we achieved the best scores for the BERT-based models, which yielded accuracy of over 90% for sentiment (3-class) classification and almost 80% for emotion (9-class) classification. We compared the results for both languages and discussed the differences. Both the accuracy and the F1-scores for Polish turned out to be slightly inferior to those for English, with the highest difference visible for BERT.



**Citation:** Zygadło, A.; Kozłowski, M.; Janicki, A. Text-Based Emotion Recognition in English and Polish for Therapeutic Chatbot. *Appl. Sci.* **2021**, *11*, 10146. <https://doi.org/10.3390/app111010146>

Academic Editor: Kyle Ke

Received: 29 September 2021

Accepted: 26 October 2021

Published: 29 October 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** human-machine interaction; chatbot; sentiment recognition; emotion recognition; Polish language; parallel text corpus; fastText; BERT; machine translation

## 1. Introduction

Chatbots and dialogue systems are entering new areas of our lives. Quite recently, they have also been introduced to psychological and psychiatric therapies. Such a system, in order to have a therapeutic conversation with a patient, must be able to correctly recognize the patient's emotional state. In some cases, it is enough just to recognize the sentiment of the patient, i.e., to detect whether the patient's utterance has a positive or a negative emotional tinge, or carries no emotion at all. In other cases, to correctly lead the therapeutic dialogue, more detailed emotion and mood recognition must be performed.

Our long-term target is to create a therapeutic dialogue system, working in Polish, able to hold empathetic conversations with patients. Most of the existing sentiment analysis approaches for Polish are opinion-based. They work either with short texts, e.g., as in message-level Twitter sentiment polarity classification, or with longer texts, e.g., when analyzing reviews of multiple aspects of restaurants or other services or goods. However, emotion-aware dialogue agents demand a different type of dataset, one that is more empathetic and multifaceted rather than just opinion-forming.

The Polish language is under-resourced in regard to annotated empathetic texts. To fill in this gap, in our work, we made an attempt to build a Polish version of the dataset, using neural machine translation and English corpora as the source texts.

In addition, we observed that in their experiments many researchers avoid taking into account the neutral emotional state, even though neutral utterances usually prevail during conversations. Since correctly distinguishing between neutral sentiment/emotion and any other emotional state is quite important in therapy, we decided to combine emotion-labeled texts with neutral ones. Next, we ran several experiments with sentiment polarity

and emotion recognition for English and Polish, comparing the results between various classifiers and both languages.

Our article is structured as follows: first, in Section 2, we briefly review the state of the art in the area of dialogue systems applied to mental health, sentiment and emotion recognition, and the existing related resources. Next, in Section 3, we describe how we created the corpora for our study. The experiments themselves are described in Section 4, followed by presentation of the results in Section 5. The article concludes with discussion of the results in Section 6 and a summary in Section 7.

## 2. Related Work

### 2.1. Conversational Agents in Mental Health

Mental disorders affect large numbers of people worldwide. To mitigate the problem of limited availability of human therapists and to develop new methods of treatment, computer-aided therapies have been designed and successfully applied in the context of mental health, including solutions based on artificial intelligence (AI) [1].

Application of AI in mental disorder therapies frequently takes the form of dialogue systems [2,3], which can be divided into two categories. The first is chatbots—virtual counselors capable of having text-based conversations with the patient, delivered, e.g., via a mobile application. Research [4–6] has shown positive impacts from chatbot-based therapy on patients with depression and anxiety. Another group of systems is the so-called embodied conversational agents (ECAs), which extend the text conversations with an animated visualization of the virtual therapist on the screen. Such systems have been applied in the treatment of depression [7] and autism spectrum disorders [8].

An important factor in human-computer interaction in therapeutic settings is the system's ability to recognize the patient's emotions and respond accordingly (*affective computing* [9]). Several affect-aware conversational systems have been developed [10,11] in the context of mental health, aiming to produce more natural and empathetic conversations; however, to the best of our knowledge, none of these were designed for the Polish language.

### 2.2. Text-Based Sentiment and Emotion Recognition

The long-term goal of our research is to develop a therapeutic chatbot capable of having a conversation in Polish. Such a system should not only be able to respond according to the user's intent and the topics mentioned, but its utterances should also be aligned with the user's emotional state. For this purpose, various text-based sentiment and emotion recognition methods have been developed.

The majority of historical approaches to sentiment analysis employed bag-of-words (BoW) representations and machine learning (ML) algorithms to build classifiers from textual data (e.g., utterances, opinions, reviews) with manually annotated sentiment polarity (e.g., positive, negative, neutral). Most studies focused on designing effective features to obtain better classification performance [12]. Snyder and Barzilay [13] analyzed the sentiment of multiple aspects of restaurant reviews, such as food and atmosphere. Several works have explored sentiment compositionality through careful engineering of features or polarity-shifting rules on syntactic structures [14]. Psycholinguistic features can be built using the large lexicons of word categories (LIWC) [15] that represent psycholinguistic processes (e.g., perceptual processes) and summary categories (e.g., word ratio), as well as part-of-speech categories (e.g., articles, verbs). Mohammad et al. [16] implemented diverse sentiment lexicons and a variety of handcrafted features.

Further progress toward understanding compositionality in tasks, such as sentiment detection, requires more complex datasets and more powerful ML models, such as deep-learning (DL) models. Socher et al. [17] introduced a new corpus—the Stanford Sentiment Treebank (SST), and a new approach based on DL—the Recursive Neural Tensor Network. SST includes fine-grained sentiment labels for 215,154 phrases in the parse trees of 11,855 sentences and presents new challenges for sentiment compositionality. The proposed method, trained on the new treebank, outperformed all previous methods.

Recent works have shown that shallow neural networks can also perform well for sentiment classification. Reference [18] presents the use of fastText word embeddings [19] as representation of words to perform the task of sentiment analysis. The results showed that the proposed approach yielded better results than many classic baseline models.

Currently, BERT (Bidirectional Encoder Representations from Transformers) is reported to be the state-of-the-art language model and has achieved amazing results in many language understanding tasks [20], including sentiment recognition. In Reference [21], the authors used the pretrained BERT model and fine-tuned it for the fine-grained sentiment-classification task on the Stanford Sentiment Treebank dataset. The proposed model performed better than complicated architectures, such as paragraphVectors, or typical recursive and convolutional neural networks. However, BERT-based models also exhibit some limitations, e.g., they have large computational and memory requirements, and the black-box model characteristics make their predictions hardly interpretable.

Deep-learning approaches, such as BERT-based models, have recently achieved state-of-the-art results [22] in the *SemEval 2018* competition’s task related to detecting affect in Tweets [23], with objectives ranging from emotion classification to emotion-intensity prediction. The competition dataset was annotated with 12 different emotions (incl. *no emotion*) for English, Spanish and Arabic in a multi-label manner. The most successful approaches during the competition in 2018 used combinations of sentence embeddings with features extracted from affective lexicons.

The interest in emotion analysis as part of cyclical competitions materialized one year later in *SemEval 2019*, in a task called *EmoContext* [24]. Its goal was to classify the emotion represented by a short informal dialogue utterance, also taking into account the preceding two turns of the dialogue. There were only four classes of emotion (*Happy*, *Sad*, *Angry*, and *Others*) and three different classification subtasks. Among the top systems, there were again examples of models leveraging both vector representations of sentences and emotion-related features.

Even though there are many English-language corpora of emotional texts, for example, Reference [13,17,25], only a few relevant resources exist for Polish. Until recently, the majority of approaches to sentiment analysis in Polish were based on lexicons, such as *plWordNet 4.0 Emo* [26,27] or the *Nencki Affective Word List (NAWL)* [28]. In recent years, several sentiment-labeled corpora have also been created. One is *PolEmo* [29], a corpus of consumer reviews from four domains: medicine, hotels, products, and schools. It contains 8216 reviews having 57,466 sentences. The corpus is labeled, both on review and sentence level, with four polarity classes: positive, neutral, negative, and ambiguous. Another example is the *HateSpeech* corpus [30], the current version of which contains over 2000 manually annotated posts crawled from the Polish public web. The posts contain various types and degrees of offensive language, expressed toward minorities (e.g., ethnic, racial). However, to the best of our knowledge, no emotion-labeled corpora exist for Polish. In addition, the current corpora contain no dialogue phrases. Our work aims to fill these gaps by creating a new corpus, suitable for the context of a chatbot.

### 2.3. Dialogue Corpora

To develop an affect-aware dialogue system, relevant conversational datasets are required. Most of the existing dialogue corpora are either domain-specific and task-oriented [31,32] or collected without full control over the content, e.g., from social media [33,34], and, therefore, are generally inappropriate in a therapeutic setting. There are, however, examples of emotionally-grounded conversational datasets for English, such as *EmpatheticDialogues* [35] and *DailyDialog* [36], which are labeled with emotions at the dialogue and utterance level, respectively.

The *DailyDialog* dataset [36] consists of daily conversations obtained through crawling websites for English learners. In total, it contains 13k dialogues, manually labeled with dialogue acts and emotions at the utterance level. The emotion-labeling scheme distinguishes six basic emotions: anger, disgust, fear, happiness, sadness, and surprise.

The emotion distribution in the dataset is highly imbalanced, with “happiness” being more than 10 times more frequent than the other emotions. There are also a large number of utterances (83% of the entire dataset) marked as representing no emotion.

The authors of Reference [35] propose a new benchmark for empathetic dialogue generation and EmpatheticDialogues (ED) itself—a novel dataset with about 25k personal dialogues. Each dialogue is grounded in a specific situation where the speaker was feeling a given emotion, with the listener responding actively. The resource consists of crowdsourced one-on-one conversations, and covers a large set of emotions in a balanced way. This dataset is larger and contains a more extensive set of emotions than many similar emotion-prediction datasets from other text domains. The authors’ experiments show that models built on this dataset are considered more empathetic by human evaluators, compared to models merely trained on large-scale Internet-crawled opinion-oriented data.

To the best of our knowledge, no similar dialogue corpora exist for Polish.

### 3. Materials and Methods

We decided to create a corpus for emotion recognition from two sources: the EmpatheticDialogues and DailyDialog datasets, previously presented in Section 2.3. Within this research, following the classification experiment described in Reference [35], we frame the problem of emotion recognition as a single dialogue turn classification. Therefore, it was sufficient for creation of the corpus for classification to take just one utterance from each of the dialogues.

#### 3.1. Extending the Corpus with Neutral Texts

The utterances in the EmpatheticDialogues corpus were collected by providing the dialogue participants with a *prompt* sentence, together with a *context* label, representing one of 32 emotional groundings in which several dialogue turns were then produced, starting from the prompt or a slightly modified version of it. In most dialogues, the emotional grounding is reflected in the first dialogue turn, but this is not always the case. Bearing that in mind, we decided to build our corpus for emotion recognition based on the prompt sentences rather than on the dialogue content itself.

Considering the planned future usage of the developed emotion detector in the context of a therapeutic chatbot, it was necessary to include neutral utterances in the classification corpus. Unfortunately, there were no examples of labeled neutral sentences in the EmpatheticDialogues dataset. Therefore, for this purpose, sentences from the DailyDialog corpus were used. For each of the experiments conducted, neutral (“no emotion”) sentences were sampled from DailyDialog data, avoiding duplicated entries, equal in number to the mean count of all the other classes in the experiment. The original corpora were split into training, validation, and test subsets by their authors, and we decided to retain this division in the resulting dataset.

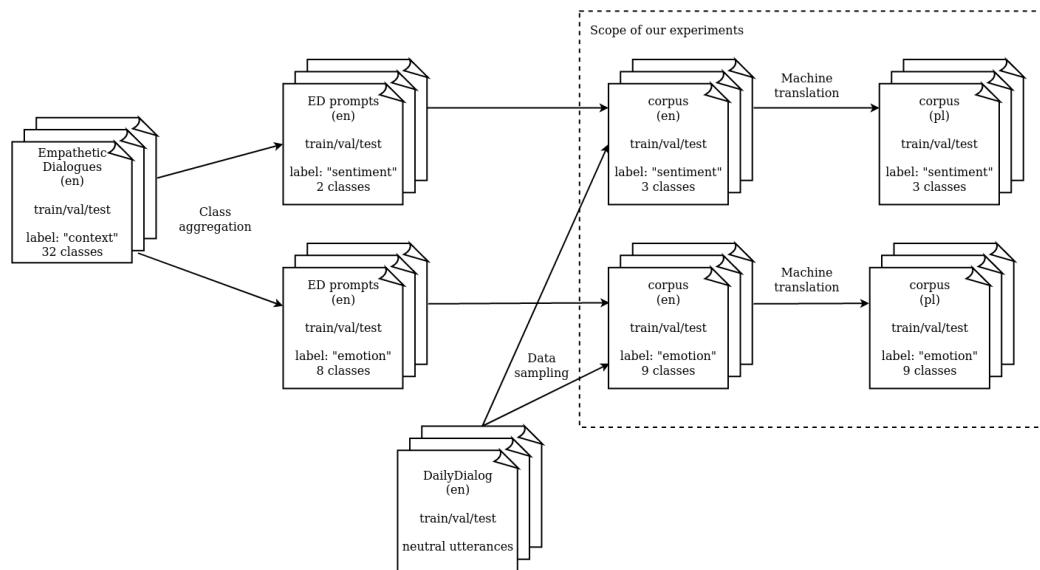
#### 3.2. Creating the Polish Corpus Using Machine Translation

Since we aim to create a sentiment and emotion classifier for a chatbot working in Polish, we faced the problem of lack of relevant dialogue datasets for the Polish language. As a cheap and fast solution, we proposed to take advantage of the available resources for English and obtain the desired Polish sentences via machine translation (MT). Such an approach has already turned out to be successful, e.g., in creating a corpus for virtual assistants [37].

We tried several available MT solutions, translating from English into Polish and observed the correctness of the translations. Eventually we chose the Google Translation API (<https://cloud.google.com/translate>, accessed on 21 May 2021), a neural MT system, as it gave the highest level of correctness among the tested MT systems.

The whole process of creating our corpus is shown in Figure 1. Eventually we created a parallel bilingual (English and Polish) corpus of emotional texts, designed to serve

experiments on sentiment and emotion recognition. We named it CORTEX, or CORpus of Translated Emotional teXts.



**Figure 1.** Process of creating corpora for emotion recognition.

## 4. Experiments

### 4.1. Models

We evaluated several approaches to emotion recognition, both simpler and more complex ones. The baseline models were the multinomial Naïve Bayes and linear Support Vector Machine classifiers trained on top of BoW representation applied to token bigrams. Both algorithms are available within the *scikit-learn* (<https://scikit-learn.org>, accessed on 21 May 2021) framework. Another model was based on the fastText algorithm, obtained from pretrained word embeddings (300-dimensional variant) available in the *fasttext* library (<https://fasttext.cc>, accessed on 21 May 2021), both for English and Polish, and also using token bigrams. For other fastText hyperparameters, we used their default values, including training for 5 epochs and a learning rate of 0.1.

The most complex approach was to fine-tune pretrained BERT<sub>BASE</sub> models for English and Polish (<https://huggingface.co/dkleczek/bert-base-polish-uncased-v1>, accessed on 21 May 2021) (the uncased variants) to the task of sequence classification. We performed fine-tuning using the AdamW optimizer with a linear learning rate decay starting from the maximum value of learning rate of  $2 \times 10^{-5}$ , preceded by a warm-up for 10% of steps. We trained the models for 4 epochs with an effective batch size of 24, and we selected the best model for each experiment based on the validation metrics obtained after each training epoch. The code for training BERT was developed with the *HuggingFace Transformers* library [38].

### 4.2. Classes of Emotions

The corpus created as a combination of EmpatheticDialogues and DailyDialog utterances contained 32 classes of emotions plus the neutral class. This number seemed unnecessarily high considering the context of a therapeutic chatbot.

For the purpose of developing emotion classification models, we introduced new levels of class aggregation. First, we excluded some of the original emotion labels (anxious, surprised, impressed, nostalgic, sentimental, anticipating), which proved to be difficult to assign, as the utterances represented a given emotion both in positive and negative situations. Such ambiguous emotions might introduce noise to the training process. Next, we grouped similar emotion classes (see Table 1), taking into account the original papers that were the inspiration for emotion inventory used in Reference [35]. This led to two experimental setups—*sentiment* (3 classes) and *emotion* (9 classes) classification.

**Table 1.** Class aggregations and corpus statistics (# sentences) for *sentiment* and *emotion* setups.

Sentiment	# Sentences Train/Val/Test	Emotion	# Sentences Train/Val/Test	Classes Used in Reference [35]
positive	5985/807/826	happiness	2391/310/328	excited/joyful/grateful/content
		confidence	1705/237/222	confident/prepared/hopeful
		other positive	1889/260/276	proud/trusting/caring/faithful
negative	8314/1133/1080	sadness	2267/322/289	sad/lonely/disappointed/devastated
		anger	1804/243/226	angry/annoyed/furious
		fear	1565/212/207	afraid/terrified/apprehensive
		guilt	1576/215/199	embarrassed/ashamed/guilty
		other negative	1102/141/159	jealous/disgusted
neutral	7149/970/953	neutral	1787/242/238	no emotion
<b>total</b>	<b>21,448/2910/2859</b>	<b>total</b>	<b>16,086/2182/2144</b>	

## 5. Results

We evaluated four different models for *sentiment* and *emotion* recognition, for each of the languages, using the developed CORTEX dataset. The numbers of sentences in individual subsets (train/val/test) are displayed in Table 1. In each experiment, we measured the values of accuracy and support-weighted F1-score (see Table 2). We conducted statistical analyses using the Wilson score interval, for the confidence level set to 90%. We assessed the confidence intervals for F1-score based on the confidence intervals for precision and recall. We also generated confusion matrices, allowing for a more detailed analysis of the results, including the models' mistakes.

**Table 2.** Results of experiments (in percentages) on sentiment and emotion recognition for English and Polish versions of the corpus for test subset. Confidence intervals given for confidence level 90%.

Experiment	Metric	Language	Classifier			
			NB	SVM	FT	BERT
Sentiment (3-class)	Accuracy	en	85.41 ± 1.08	86.99 ± 1.03	88.07 ± 0.99	93.74 ± 0.74
		pl	83.00 ± 1.15	84.89 ± 1.10	85.59 ± 1.08	92.24 ± 0.82
	F1-score	en	85.44 ± 1.05	86.89 ± 1.02	88.00 ± 0.97	93.75 ± 0.71
		pl	83.08 ± 1.12	84.75 ± 1.08	85.43 ± 1.06	92.26 ± 0.79
Emotion (9-class)	Accuracy	en	63.29 ± 1.71	65.11 ± 1.69	68.42 ± 1.65	78.96 ± 1.44
		pl	62.27 ± 1.72	63.43 ± 1.71	65.25 ± 1.69	75.19 ± 1.53
	F1-score	en	63.06 ± 1.70	64.86 ± 1.67	68.33 ± 1.63	79.08 ± 1.40
		pl	62.06 ± 1.71	63.11 ± 1.70	65.01 ± 1.67	75.15 ± 1.49

As seen in the table above, in both experimental setups (*sentiment* and *emotion*), BERT models outperformed the simpler methods, reaching over 90% accuracy for sentiment classification and almost 80% for emotion classification. The F1-score yielded similar values. The results for the test subset were usually slightly inferior to those for the validation subset; however, the difference was not high. The results obtained for Polish were by a few relative % worse than for English, especially in the case of BERT for *emotion* recognition. Unsurprisingly, for the 3-class scenario, evaluation metrics reached much higher values than for the 9-class scenario. The baseline models achieved visibly worse results, with SVM being slightly better than Naïve Bayes. These classifiers were based on the BoW representation, and, therefore, were not able to express semantic similarity between tokens in a proper way.

We observed a slight improvement for the fastText-based classifier applied to token bigrams. In this algorithm, the embeddings were created without the context of the entire input sequence, and this might be why BERT outperformed it by a large margin.

The scores for the Polish corpora were worse than for their English counterparts. This difference was significant for the more complex models. Apart from the degraded

quality of the translated sentences, this might also have been caused by the quality of the underlying pretrained embeddings. Nevertheless, as presented in Table 3, both English and Polish models learned to distinguish neutral sentences from emotional ones (the F1-score for the neutral class was around 97%). Positive and negative polarity predictions reached between 88% and 92%, with slightly higher per-class scores for negative polarity. It is worth noting that only the BERT<sub>BASE</sub> (uncased) variants were evaluated, while plenty of other contextual embedding models are available.

**Table 3.** Evaluation metrics (in percentages) for sentiment classification with BERT model for test subsets, for both languages. Confidence intervals given for confidence level 90%.

Language	Sentiment	Precision	Recall	F1-Score
English	positive	89.49 ± 0.94	92.74 ± 0.80	91.08 ± 0.87
	negative	93.54 ± 0.75	91.20 ± 0.87	92.36 ± 0.81
	neutral	97.79 ± 0.45	97.48 ± 0.48	97.64 ± 0.47
Polish	positive	87.80 ± 1.00	88.86 ± 0.96	88.33 ± 0.98
	negative	90.68 ± 0.89	91.94 ± 0.83	91.31 ± 0.86
	neutral	98.06 ± 0.42	95.49 ± 0.64	96.76 ± 0.53

## 6. Discussion

The envisioned study objectives have been met: we have created and tested a sentiment (3-class) and emotion (9-class) text-based classification engine for a therapeutic dialogue system, working in Polish. To achieve this, we had to create our own emotion-labeled corpus, which we generated using a neural MT system and two source English corpora. For sentiment and emotion recognition, we employed the state-of-the-art deep-learning classifier based on the BERT model, which outperformed the classic models, such as Naïve Bayes or Support Vector Machines.

We analyzed the misclassifications made by the best model (BERT) in more detail by looking at the examples related to the highest values from the confusion matrix (Table 4), especially the cases when a positive emotion label was confused with a negative emotion prediction and vice versa. The most problematic class was *other\_positive*, as it was quite frequently predicted for sentences labeled with negative emotions, such as *anger*, *sadness*, and *other\_negative*. The models for both languages did well in distinguishing between neutral and emotional texts; we obtained high F1-scores for the neutral class: 97.6% and 96.8% for English and Polish, respectively.

**Table 4.** Confusion matrix for emotion classification in Polish with BERT model for test subset. Confidence intervals given for confidence level 90%.

Emotion	happin.	conf.	o_pos	anger	fear	sadness	guilt	o_neg	neutral	F1-Score [%]
happiness	247	15	27	4	5	10	8	5	7	75.08% ± 1.53
confidence	24	158	15	2	7	7	3	2	4	74.70% ± 1.54
other_pos	39	16	180	5	6	8	10	10	2	67.04% ± 1.67
anger	3	0	7	159	12	13	10	18	4	70.04% ± 1.62
fear	0	4	4	8	171	10	7	3	0	80.28% ± 1.41
sadness	10	0	8	17	9	215	11	14	5	75.84% ± 1.52
guilt	0	3	7	15	2	10	148	13	1	73.45% ± 1.56
other_neg	3	1	11	18	3	5	7	111	0	66.07% ± 1.68
neutral	4	4	2	0	4	0	0	1	223	92.15% ± 0.95

We can explain some of the failed predictions as errors in the translation, others by prompt difficulty, e.g., the emotion was not reflected in the prompt itself (example: *I have some friends who are traveling all over Europe* taken from a dialogue labeled *jealous*) or multiple emotions were present in the utterance (example: *I recently said goodbye to a good friend for a*

while. I love her!). Sometimes the label itself was just wrong, e.g., some of the neutral texts from DailyDialog seemed to be missing emotional labels (*I'm happy with that price*—labeled with *no emotion*—for which the model predicted *happiness*).

One of the limitations of our study is the accuracy of employed MT. Translation errors are the inevitable cost of our fast method of creating an emotion-labeled corpus for a new language. To assess the level of this inaccuracy, we manually verified a sample of our corpus. We found that about 10% of cases contained minor translation mistakes. Nevertheless, we observed that only about one-fourth of these might have an impact on the emotion category.

Considering how the dataset was obtained (machine translation from English, noisy labels), we consider the experiment results satisfactory. In the future, we plan to manually go through the developed corpus, fix the translation errors and the label mismatch where necessary, and check whether this improves the performance of our emotion-classification models.

## 7. Conclusions

In this article, we presented the results of our experiments on sentiment polarity and emotion recognition for English and Polish texts, aiming to work in the context of a therapeutic chatbot. We extended the existing language resources by adding samples of neutral texts to an existing English corpus. Next, we created a Polish version of the English database using neural machine translation. We used the corpus created in this way, which we named CORTEX, for experiments on sentiment and emotion classification. To show statistical significance, we calculated the Wilson score interval for each evaluation metric.

The results obtained were satisfactory: the best scores were achieved for the BERT-based classifiers, where accuracy of over 90% was achieved for sentiment (3-class) classification and almost 80% for emotion (9-class) classification. The results for Polish always turned out inferior to those for English, which might be caused either by imperfections in the MT process, or by the nature of the Polish language itself, as it is characterized by a more complex grammar and morphology. Exact research on this topic will be the subject of future work.

Our novel contributions presented in this article are as follows:

- From existing resources, we created a new dataset containing empathetic utterances in English, which were annotated with nine emotion classes, including neutral texts.
- Using neural machine translation, we created a Polish version of the above database, thus filling a gap in text resources for the Polish language. The two language versions of the database formed a new parallel corpus, named CORTEX.
- We ran a series of experiments with sentiment polarity and emotion classification, establishing that the BERT-based classifier is currently the best method. Thus, we set a baseline for potential future researchers.
- We showed the difference in classification efficacy for English and Polish and discussed possible explanations for this.

We made CORTEX, the developed dataset, available to the research community at <https://github.com/azygadlo/CORTEX>, accessed on 21 May 2021 and encourage researchers to use it for future experiments. We believe that this will help in designing better, more empathetic chatbots and dialogue systems, both for English and Polish. We also strongly encourage the creation of new versions of our database by extending it with next language versions.

**Author Contributions:** Conceptualization, A.Z., M.K. and A.J.; methodology, A.Z., M.K. and A.J.; software, A.Z.; validation, A.Z.; investigation, A.Z., M.K. and A.J.; data curation, A.Z.; writing—original draft preparation, A.Z., M.K. and A.J.; writing—review and editing, A.Z., M.K. and A.J.; visualization, A.Z.; supervision, A.J.; project administration, A.J.; funding acquisition, A.Z., M.K. and A.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** Research was funded by the Center for Priority Research Area Artificial Intelligence and Robotics of the Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) program.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** CORTEX is freely available at <https://github.com/azygadlo/CORTEX>, accessed on 21 May 2021.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Luxton, D. *Artificial Intelligence in Behavioral and Mental Health Care*; Academic Press: Cambridge, MA, USA, 2015; pp. 1–293. <https://doi.org/10.1016/C2013-0-12824-3>.
2. Abd-alrazaq, A.A.; Alajlani, M.; Alalwan, A.A.; Bewick, B.M.; Gardner, P.; Househ, M. An overview of the features of chatbots in mental health: A scoping review. *Int. J. Med. Inform.* **2019**, *132*, 103978. <https://doi.org/10.1016/j.ijmedinf.2019.103978>.
3. Laranjo, L.; Dunn, A.; Tong, H.L.; Kocaballi, A.; Chen, J.A.; Bashir, R.; Surian, D.; Gallego, B.; Magrabi, F.; Lau, A.; et al. Conversational agents in healthcare: A systematic review. *J. Am. Med. Inform. Assoc.* **2018**, *25*, 1248–1258.
4. Fitzpatrick, K.K.; Darcy, A.; Vierhile, M. Delivering Cognitive Behavior Therapy to Young Adults With Symptoms of Depression and Anxiety Using a Fully Automated Conversational Agent (Woebot): A Randomized Controlled Trial. *JMIR Ment Health* **2017**, *4*, e19. doi:10.2196/mental.7785.
5. Fulmer, R.; Joerin, A.; Gentile, B.; Lakerink, L.; Rauws, M. Using Psychological Artificial Intelligence (Tess) to Relieve Symptoms of Depression and Anxiety: Randomized Controlled Trial. *JMIR Ment Health* **2018**, *5*, e64. doi:10.2196/mental.9782.
6. Inkster, B.; Sarda, S.; Subramanian, V. An Empathy-Driven, Conversational Artificial Intelligence Agent (Wysa) for Digital Mental Well-Being: Real-World Data Evaluation Mixed-Methods Study. *JMIR Mhealth Uhealth* **2018**, *6*, e12106. doi:10.2196/12106.
7. Ring, L.; Bickmore, T.; Pedrelli, P. An Affectively Aware Virtual Therapist for Depression Counseling. In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2016) Workshop on Computing and Mental Health, San Jose, CA, USA, 7–12 May 2016; p. 01951-12.
8. Tanaka, H.; Negoro, H.; Iwasaka, H.; Nakamura, S. Embodied conversational agents for multimodal automated social skills training in people with autism spectrum disorders. *PLoS ONE* **2017**, *12*, e0182151.
9. Picard, R.W. *Affective Computing*; MIT Press: Cambridge, MA, USA, 1997.
10. Ghandeharioun, A.; McDuff, D.; Czerwinski, M.; Rowan, K. EMMA: An Emotion-Aware Wellbeing Chatbot. In Proceedings of the 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII), Cambridge, UK, 3–6 September 2019; pp. 1–7. doi:10.1109/ACII.2019.8925455.
11. Miner, A.; Chow, A.; Adler, S.; Zaitsev, I.; Tero, P.; Darcy, A.; Paepcke, A. Conversational Agents and Mental Health: Theory-Informed Assessment of Language and Affect. In Proceedings of the 4th International Conference on Human Agent Interaction (HAI 2016), Singapore, 4–7 October 2016; pp. 123–130. doi:10.1145/2974804.2974820.
12. Pang, B.; Lee, L. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.* **2008**, *2*, 1–135.
13. Snyder, B.; Barzilay, R. Multiple aspect ranking using the good grief algorithm. In Proceedings of the Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Main Conference, Rochester, NY, USA, 22–27 April 2007; pp. 300–307.
14. Nakagawa, T.; Inui, K.; Kurohashi, S. Dependency tree-based sentiment classification using CRFs with hidden variables. In Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, CA, USA, 2–4 June 2010; pp. 786–794.
15. Pennebaker, J.W.; Francis, M.E.; Booth, R.J. Linguistic inquiry and word count: LIWC 2001. *Mahway Lawrence Erlbaum Assoc.* **2001**, *71*, 2001.
16. Mohammad, S.M.; Turney, P.D. Crowdsourcing a word–emotion association lexicon. *Comput. Intell.* **2013**, *29*, 436–465.
17. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.
18. Santos, I.; Nedjah, N.; de Macedo Mourelle, L. Sentiment analysis using convolutional neural network with fastText embeddings. In Proceedings of the 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Arequipa, Peru, 8–10 November 2017; pp. 1–5.
19. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146.
20. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
21. Munikar, M.; Shakya, S.; Shrestha, A. Fine-grained sentiment classification using BERT. In Proceedings of the 2019 Artificial Intelligence for Transforming Business and Society (AITB), Kathmandu, Nepal, 5 November 2019; Volume 1, pp. 1–5.

22. Alhuzali, H.; Ananiadou, S. SpanEmo: Casting Multi-label Emotion Classification as Span-prediction. *arXiv* **2021**, arXiv:2101.10038.
23. Mohammad, S.; Bravo-Marquez, F.; Salameh, M.; Kiritchenko, S. SemEval-2018 Task 1: Affect in Tweets. In Proceedings of the 12th International Workshop on Semantic Evaluation, New Orleans, LA, USA, 5–6 June 2018; Association for Computational Linguistics: New Orleans, LA, USA, 2018; pp. 1–17, doi:10.18653/v1/S18-1001.
24. Chatterjee, A.; Narahari, K.; Joshi, M.; Agrawal, P. SemEval-2019 Task 3: EmoContext Contextual Emotion Detection in Text. In Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval 2019), Minneapolis, MN, USA, 6–7 June 2019; ACL: Minneapolis, MN, USA, 2019; pp. 39–48, doi:10.18653/v1/S19-2005.
25. Pang, B.; Lee, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv* **2004**, arXiv:0409058.
26. Zaśko-Zielińska, M.; Piasecki, M.; Szpakowicz, S. A Large Wordnet-based Sentiment Lexicon for Polish. In Proceedings of the International Conference Recent Advances in Natural Language Processing, Hissar, Bulgaria, 7–9 September 2015; INCOMA Ltd. Shoumen, BULGARIA: Hissar, Bulgaria, 2015; pp. 721–730.
27. Kocoń, J.; Janz, A.; Piasecki, M. Classifier-based Polarity Propagation in a WordNet. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018; European Language Resources Association (ELRA): Miyazaki, Japan, 2018.
28. Riegel, M.; Wierzba, M.; Wypych, M.; Żurawski, L.; Jednoróg, K.; Grabowska, A.; Marchewka, A. Nencki Affective Word List (NAWL): The cultural adaptation of the Berlin Affective Word List–Reloaded (BAWL-R) for Polish. *Behav. Res. Methods* **2015**, *47*, 1222–1236, doi:10.3758/s13428-014-0552-1.
29. Kocoń, J.; Miłkowski, P.; Zaśko-Zielińska, M. Multi-Level Sentiment Analysis of PolEmo 2.0: Extended Corpus of Multi-Domain Consumer Reviews. In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), Hong Kong, China, 3–4 November 2019; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 980–991, doi:10.18653/v1/K19-1092.
30. Troszyński, M.; Wawer, A. Czy komputer rozpozna hejtera? Wykorzystanie uczenia maszynowego (ML) w jakościowej analizie danych. [Can a Computer Recognize Hate Speech? Machine Learning (ML) in Qualitative Data Analysis]. *PrzegląD Socjal. Jakościowej* **2017**, *XIII*, 62–80.
31. Budzianowski, P.; Wen, T.H.; Tseng, B.H.; Casanueva, I.; Ultes, S.; Ramadan, O.; Gašić, M. MultiWOZ—A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 2–4 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 5016–5026, doi:10.18653/v1/D18-1547.
32. Hemphill, C.T.; Godfrey, J.J.; Doddington, G.R. The ATIS Spoken Language Systems Pilot Corpus. In Proceedings of the Speech and Natural Language: Proceedings of a Workshop, Hidden Valley, PA, USA, 24–27 June 1990.
33. Henderson, M.; Budzianowski, P.; Casanueva, I.; Coope, S.; Gerz, D.; Kumar, G.; Mrkšić, N.; Spithourakis, G.; Su, P.H.; Vulić, I.; et al. A Repository of Conversational Datasets. *arXiv* **2019**, arXiv:1904.06472.
34. Ritter, A.; Cherry, C.; Dolan, W.B. Data-Driven Response Generation in Social Media. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11), Edinburgh, UK, 27–31 July 2011; Association for Computational Linguistics: Stroudsburg, PA, USA, 2011; pp. 583–593.
35. Rashkin, H.; Smith, E.M.; Li, M.; Boureau, Y.L. Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv* **2018**, arXiv:1811.00207.
36. Li, Y.; Su, H.; Shen, X.; Li, W.; Cao, Z.; Niu, S. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Taipei, Taiwan, 27 November–1 December 2017; Asian Federation of Natural Language Processing: Taipei, Taiwan, 2017; pp. 986–995.
37. Sowański, M.; Janicki, A. Leyzer: A Dataset for Multilingual Virtual Assistants. In *Lecture Notes in Computer Science, Proceedings of the Conference on Text, Speech, and Dialogue (TSD2020)*, Brno, Czech Republic, 8–11 September 2020; Sojka, P., Kopeček, I., Pala, K., Horák, A., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 477–486.
38. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; pp. 38–45.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/363516748>

# Teaching AI when to care about gender

Article · August 2022

---

CITATIONS

0

READS

10

1 author:



James Powell

Los Alamos National Laboratory

71 PUBLICATIONS 181 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Applications of Neural Networks [View project](#)



Applied NLP [View project](#)

[Mission](#)[Editorial Committee](#)[Process and Structure](#)[Code4Lib](#) **Search**

Issue 54, 2022-08-29

## Teaching AI when to care about gender

*Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) concerned with solving language tasks by modeling large amounts of textual data. Some NLP techniques use word embeddings which are semantic models where machine learning (ML) is used to learn to cluster semantically related words by learning about word co-occurrences in the original training text. Unfortunately, these models tend to reflect or even exaggerate biases that are present in the training corpus. Here we describe the Word Embedding Navigator (WEN), which is a tool for exploring word embedding models. We examine a specific potential use case for this tool: interactive discovery and neutralization of gender bias in word embedding models, and compare this human-in-the-loop approach to reducing bias in word embeddings with a debiasing post-processing technique.*

by James Powell ([0000-0002-3517-7485](#)), Kari Sentz ([0000-0002-1530-1952](#)), Elizabeth Moyer ([0000-0003-1604-6049](#)), Martin Klein ([0000-0003-0130-2097](#))

### Introduction

In the 1800s, the author Mary Ann Evans wrote under the pseudonym George Eliot due to gender bias and widely embraced stereotypes about female authors. Due to the recent pandemic, online video conferencing replaced face-to-face interactions. In this isolated and impersonal space, many chose to overtly assert gender identity by specifying their preferred personal pronouns. Today we are now much more inclined to embrace our gender identity, both personally and professionally. But the passage of time does not erase the past, for which we have a vast digitized written record. So we face a new dilemma, as we become more dependent upon machine learning in our daily lives, we run the risk of unseen bias having unforeseen consequences. So we need to neutralize biases, such as those associated with gender that are perpetuated in machine learning models. To address this issue, the emerging consensus suggests we need to inspect and adjust for bias that has made its way into ML models. In other words, we ought to adjust our models, rather than our stories.

### What are word embeddings?

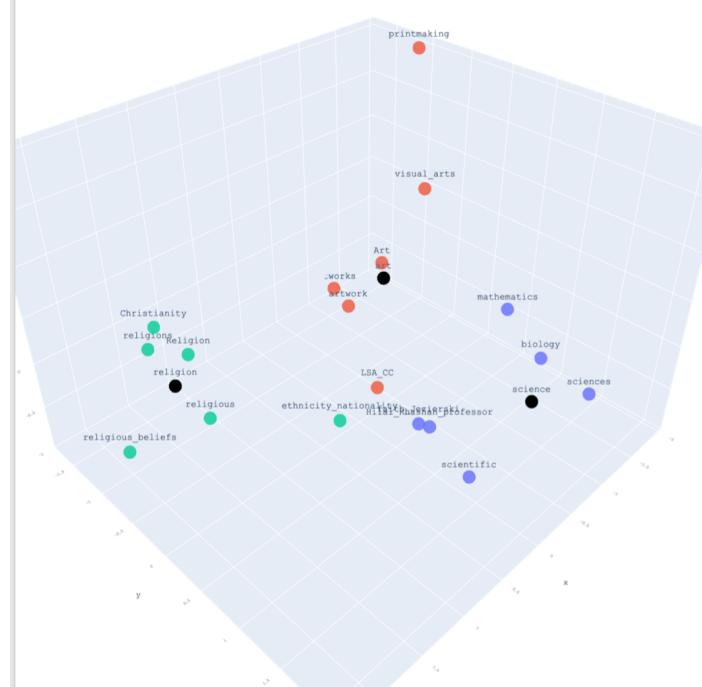
Word embedding models encode semantic information about words that they learn based on the contexts of those words in unlabeled (raw) text. There are various ways to create word embeddings, but the results are typically a word matrix of 50-300 numeric values per word that encode information about the context of each word in the corpus. These values represent the location and magnitude of a vector describing each word, which can be used to measure proximity and distance among terms in the word embedding space. The two most common approaches for generating word embeddings are [word2vec](#) (Mikolov et al. 2013) and [Global vectors for word representation \(GloVe\)](#) (Pennington et al. 2014). GloVe generates word vectors from a large matrix that contains counts of word co-occurrences across the entire training corpus. Word2vec generates a model of co-occurrence patterns in text which is built up as it slides a fixed length word “window” over the sequence of text from the corpus. This creates a model that can predict a word or a sequence of words given another word, or words, as input. The model is encoded as a list of words in the corpus, and a sequence of numbers (a vector) that is an approximate numeric representation of word co-occurrences found in the corpus. Vectors that represent similar words will have endpoints that are closer to one another (Figure 1). These vectors can be evaluated to determine just how similar two words are by using a metric such as cosine distance, which is a simple geometric measurement of the angle between two word vectors. When this angle is small, it means that two words are semantically (or in some cases, syntactically) similar.

**Figure 1.** Principal component plots for the terms art, science, and religion showing their nearest neighbors in the New York Times corpus at left, in GoogleNews at right.

Word embeddings have characteristics which make them suitable for both intrinsic tasks such as measuring word similarity, and extrinsic tasks by serving as input to Machine Learning (ML) pipelines designed to solve NLP problems (Whitaker et al. 2019). Extrinsic NLP tasks include predicting whether a text segment is positive or negative (sentiment analysis), assigning a category to a document (such as a subject heading), or determining whether one sentence logically follows another (entailment). One way to solve these kinds of problems is via supervised ML. Supervised ML requires labeled training data, that is, explicit examples of data that represent learnable patterns. Labeled training data for automated sentiment analysis might consist of movie review texts labeled as positive or negative. Given this training data, an ML algorithm can learn the patterns that constitute positive and negative sentiment (a good or bad movie review). One type of ML algorithm is a neural network. Neural networks are written to learn patterns in data, rather than to solve a specific problem. Given large amounts of training data, a neural network can learn a model that can make predictions about new, previously unseen data. Word embeddings can be used to speed up training of neural networks when the training data contains text, because word embedding models already contain useful information about text, such as which words have similar meanings. Using a pretrained model to support training of another model is referred to as transfer learning, because it enables re-use of an existing ML model to facilitate training of another model for a different task. Transfer learning also speeds up training

of new predictive models since the neural network no longer needs to relearn everything from scratch.

For many deep learning/NLP tasks, it is standard practice to download pre-trained models such as GoogleNews, Common Crawl, and wikipedia word embeddings models and use them for deep learning tasks. Since pretrained models are trained on extremely large corpora, they often contain superior word co-occurrence vectors to those that would be found in a smaller model. Large word embedding models work well if there is enough overlap between the vocabulary of the model and the vocabulary of the text of the local training corpus. If there is a significant mismatch, where there are a lot of vocabulary words not found in the embedding model, then large word embedding models are less beneficial. In those cases, researchers will train local word embedding models. This might be done for example in cases where an embedding model for the native language of the text does not exist, when the vocabulary used in the corpus is highly specialized, such as is the case with genre specific text or scientific publications, or if they want to work directly with a particular word embedding model or set of models, as is the case with diachronic text analysis. Diachronic text analysis involves using word embeddings that include a time dimension, allowing for measurement of change among words over time. The DUKWeb project (Tsakalidis et al. 2021) is a large-scale effort to generate pre-trained diachronic word embeddings for the contents of websites in the .uk subdomain retrieved from the Internet Archive, allowing researchers to study the changes in relationships and meaning among words in this corpus spanning 1996-2013.



## When word embeddings go wrong

The [Common Crawl](#) [1] word embedding model is trained on one of the largest corpora that is publicly available. The corpus consists of 2.96 billion active and defunct web pages which have been collected since 2008. A 2022 study entitled “[Based on billions of words on the internet, people = men](#)” (Bailey et al. 2022), found that in a word embedding model trained on *Common Crawl*, the cosine distance between word embedding vectors for “men” and “people” was smaller than the cosine distance for embedding vectors for “women” and “people.” Validation of this hypothesis inspired a second line of inquiry based on a follow-on hypothesis that, given the model’s conflation of “people” with “men”, gender stereotype associations in this model would be asymmetric. They also found evidence to support this. Their results showed that women were more likely to be associated with gender stereotypical language than men.

In 2016, a group of researchers wrote a [landmark paper](#) (Bolukbasi et al. 2016) about the prevalence of gender stereotypes in large word embedding models, and proposed techniques for addressing it. They first demonstrated this problem via analogies mapped to simple vector math, showing that many common gender stereotypes were present in word embeddings trained on the Google News corpus. They enlisted crowdsourcing to identify two sets of 100 pairs of words that could be used to identify definitional (e.g. she, grandmother) and stereotypical (she, secretary) gender associations. Next, they used ten of these gender pairs to identify what is in essence a direction for gender.

Their concept that gender might be associated with a direction in a word embedding model inspired them to propose several debiasing techniques for word embeddings. In one approach, they use gender-specific words to find a vector representing a gender direction common to the terms. Using this “gender” vector, they identify a second vector that is orthogonal to it. They then perform debiasing by projecting (move) gendered words onto this orthogonal vector.

This eliminates the gender direction from the terms, which they suggest results in the neutralization of gender in these word embedding vectors. This causes words that previously had a strong association with one gender to be equidistant from gender words. For example after the adjustment, “secretary” would be the same distance from “he” as it is from “she”. Their second approach aims to preserve gender associations while reducing bias. For example, this approach ensures that “grandmother” retains a relationship to “female” gender, and “grandfather” to male gender. It then adjusts terms that might have a gender association learned from bias in the corpus. This might cause the terms “hiking” and “baking” to be adjusted so that they are equidistant from “grandmother” and “grandfather.” This is a much simplified description of their work, and this paper is definitely worth a read in its entirety.

It is important to note the central role of human judgment in the research described above. The authors are careful to note that human judgment plays an important role in their approach to identifying bias and debiasing word embedding models. They broadly observe that “gender associations vary by culture and person.” They also discuss at length the process of manually creating gender word pairs suggesting that “the choice of words is subjective and ideally should be customized to the application at hand.” We will revisit these issues shortly.

Gender bias has historically had a significant impact on many aspects of life, so the identification of gender bias in data that is so central to many machine learning algorithms, and the idea that this bias could possibly be neutralized was an extremely important step for the field. This opened the door for the possibility that other forms of social bias (Garg et al. 2018) in this data could be identified and mitigated. But social biases are not the only problems that can manifest in word embeddings. Fixed word embedding models trained on large corpora will inevitably encounter words that have multiple meanings, or senses. This results in a diffuse representation that does not represent any sense of the word very well. For example, the word “bank” has several distinct uses, including “financial institution” and “the land adjacent to a river.” If a training corpus includes references to both senses of “bank”, then the resulting embedding vector will represent the merger of what are effectively two semantically distinct words represented by the same string of characters, which have different word neighbors and usages. In other cases, the quality of some word embedding vectors may suffer due to a paucity of examples in the text. Sometimes an embedding model may encode cultural biases, or be heavily influenced by false or misleading information sources, or it may skew toward particular opinions, political positions, or scientific disciplines.

Word embeddings may also have problems learning consistently meaningful representations for distinct scientific disciplines, particularly for terms that span multiple disciplines. A scientific corpus skewed towards physics and other hard sciences will generate better representations for terminology in those fields, but will offer little to no meaningful semantic representations for other fields such as sociology or linguistics. Nor would it be useful for identifying important co-occurrences that might be indicative of multidisciplinary efforts. This problem, perhaps obviously, is more likely to occur for word embeddings generated for a scientifically focused corpus, and it exemplifies the tradeoff between capturing a good representation for a technical vocabulary over modeling science more broadly or depending on a large-scale word embedding model.

Digital libraries provide wide ranging access to intellectual and cultural artifacts. Machine learning models can be used to support many features of digital libraries. They can be used to suggest topics for documents in a corpus, recommend or summarize content, perform automated text classification, and support query formulation and offer search term suggestions, to name a few. If any of these features depends on a model that has problems such as encoded bias, incomplete semantic representations, or term meanings affected by skewed points of view; it could potentially produce unpredictable or misleading or even offensive results. This could be particularly harmful if, for example, some type of bias affects the perception or distribution of historical or culturally sensitive content, or if it results in the inadvertent censoring of certain ideas due to current political sensibilities. It is also important to note that digital libraries can be the source of corpora used to produce word embedding models for various tasks, treating library collections as data (Padilla et al. 2019). Collection curators are in a particularly good position to identify and mitigate problems with word embeddings trained on their corpora if they can be empowered to do so.

## The Word Embeddings Navigator

---

We believe that one of the best ways for a human to “explain” bias to a word embedding model is to enable them to inspect that model and give them the ability to iteratively perform targeted adjustments to vectors in that model. The goal of explanation “is to fill in the gap between his audience’s knowledge or beliefs about some phenomena and what [they take] to be the actual state of affairs” [2]. With that goal in mind, we introduce the *Word Embeddings Navigator (WEN)*, an interactive Web application for exploring and modifying word embedding models. *WEN* facilitates human-in-the-loop interaction with word embedding models to enable iterative query and adjustment of term vectors. Although there are other word embedding visualization tools, including *Conceptvector* (Park et al. 2017), *Embedding Projector* (Smilkov et al. 2016), *vec2graph* (Katracheva et al. 2019), and *WordBias*, a tool which was specifically designed to facilitate bias detection through visualizations (Ghai et al. 2021); we believe this is the first tool that combines visual word embedding exploration with the ability for a user to interactively adjust individual word vectors. In the following sections we will illustrate how this works, describe some of the technical details of the system, and evaluate how it performs in comparison to post-processing debiasing.

*WEN* was originally conceived of as a tool for exploring and adjusting temporally aligned word embedding models. For example, figure 2 shows a *WEN*-generated heatmap visualization of terms near the word “mars” in temporal snapshots of a scientific corpus. *WEN* was a product of an internally funded research project (Senzt et al. 2019). A broader goal of the project was to support discovery of latent knowledge stored in an embedding model (Tshitoyan et al. 2019). Latent knowledge exists in all corpora but for scientific corpora that span significant time periods within a particular field, there are often hints of discoveries that have yet to be formally declared, and these can be represented by similar contexts that occur at different points in time. The ability to change word embedding vectors was added when it became apparent that small, locally trained models failed to capture relationships that were immediately obvious to subject matter experts, but were not reflected in the source training data. Adding a mechanism to change individual embedding vectors fit nicely into the application since those changes would be made to the embedding model in memory, and thus manifested in subsequent user queries. We envisioned that users would make these adjustments based on specialized knowledge such as that possessed by a subject matter expert.

## How WEN works

---

*WEN* is designed to allow for interactive exploration and iterative adjustment of word2vec word embedding models or other embedding models, such as GloVe embeddings, that can be converted to word2vec format. It emphasizes a four step approach: “search – navigate – visualize – adjust”. All four phases are in service of the goal of surfacing quality issues in word embeddings and making it possible to address these issues directly and immediately within the model. We implemented a prototype of this concept, at the core of which is a capability we refer to as interactive refitting. The prototype consists of the following components:

- A search interface for providing one or more search terms
- A textual result interface that presents most similar terms from the word embedding model
- Links to explore these results via three Web visualization tools

- A mechanism allowing users to identify and specify target for adjustment via refitting
- A Web services wrapper for our implementations of the refitting objective
- An update function that modifies selected word2vec embedding vectors and stores them in the live model

The embedding navigator tool makes extensive use of the `gensim` [3] python library for natural language processing. An instance of the navigator can be configured to load an arbitrary number of related or unrelated word embeddings in word2vec binary or text format, only limited by system memory. At launch time, navigator loads a configuration file, and as a Flask application, it then iterates through file system references to the embedding files and loads them into memory. Since this tool allows users to update individual word embedding vectors, we load the full word embedding model, rather than just a dictionary of words and their vectors, which would have a smaller memory footprint but would not allow model updates. At run time, `WEN` parses its configuration file to discover which models to load, and there is additional information provided to support the user interface as illustrated by this JSON metadata excerpt:

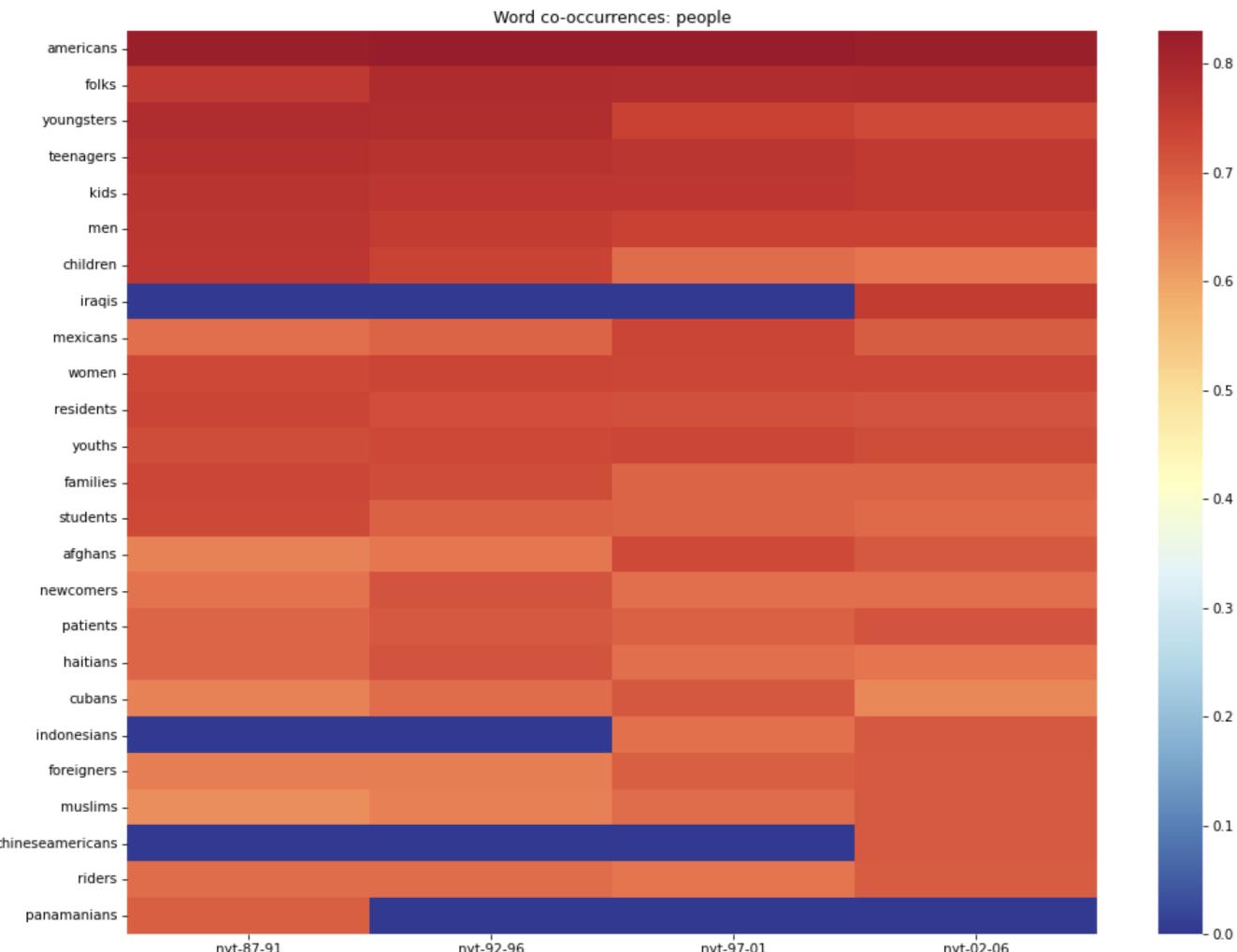
```
1 | {
2 |   'name': 'NYT 1987-1991',
3 |   'filename': 'NYT-1987-1991.model',
4 |   'label': 'nyt-87-91'
5 | }
```

The name is the full corpus name to be presented to the user, filename is the primary word2vec binary model file to be loaded at runtime, and the model label is used in the UI to provide short labels for interface form elements and various visualizations. A serialized model is loaded from the filesystem using a call to `gensim's word2Vec`, e.g.:

```
1 | this_model = Word2Vec.load(models['filename'])
```

where `models['filename']` corresponds to a file containing word2vec embeddings.

Metadata about the loaded models is utilized throughout the interface. Flask templates ensure that the user can query any or all of the embedding models depending on what aspect of the embeddings they are interested in. Since `WEN` was initially conceived of as a tool to explore temporal slices of word embeddings, there are some facilities for visualizing the behavior of a group of words associated with a query across a set of embeddings, usually a temporally aligned collection of models. A sankey diagram illustrates the changes in similarity to the target word in each instance of the loaded embeddings. This works for temporally aligned and unaligned word embeddings as well. It can also be used to view non-temporal shifts among a set of word embeddings.



**Figure 2.** WEN-generated heatmap for terms most similar to “people” across New York Times word embedding models spanning 1987-2006.

## WEN use cases

The search interface allows for several types of queries (Figure 3). The basic query is for a word or phrase in a single user selectable word embedding model. The query itself is performed using the *gensim* “most\_similar” method of the word embedding model. It computes the cosine similarity of the query word’s vector to all other word vectors in the target embedding space, and returns a list of entries that are the most similar to the query term. Somewhat counter intuitively, the most\_similar method can also be used to perform a query that involves subtracting the vector of one term from another and then finding the most similar entries to the resulting vector. We implement this as the difference query.

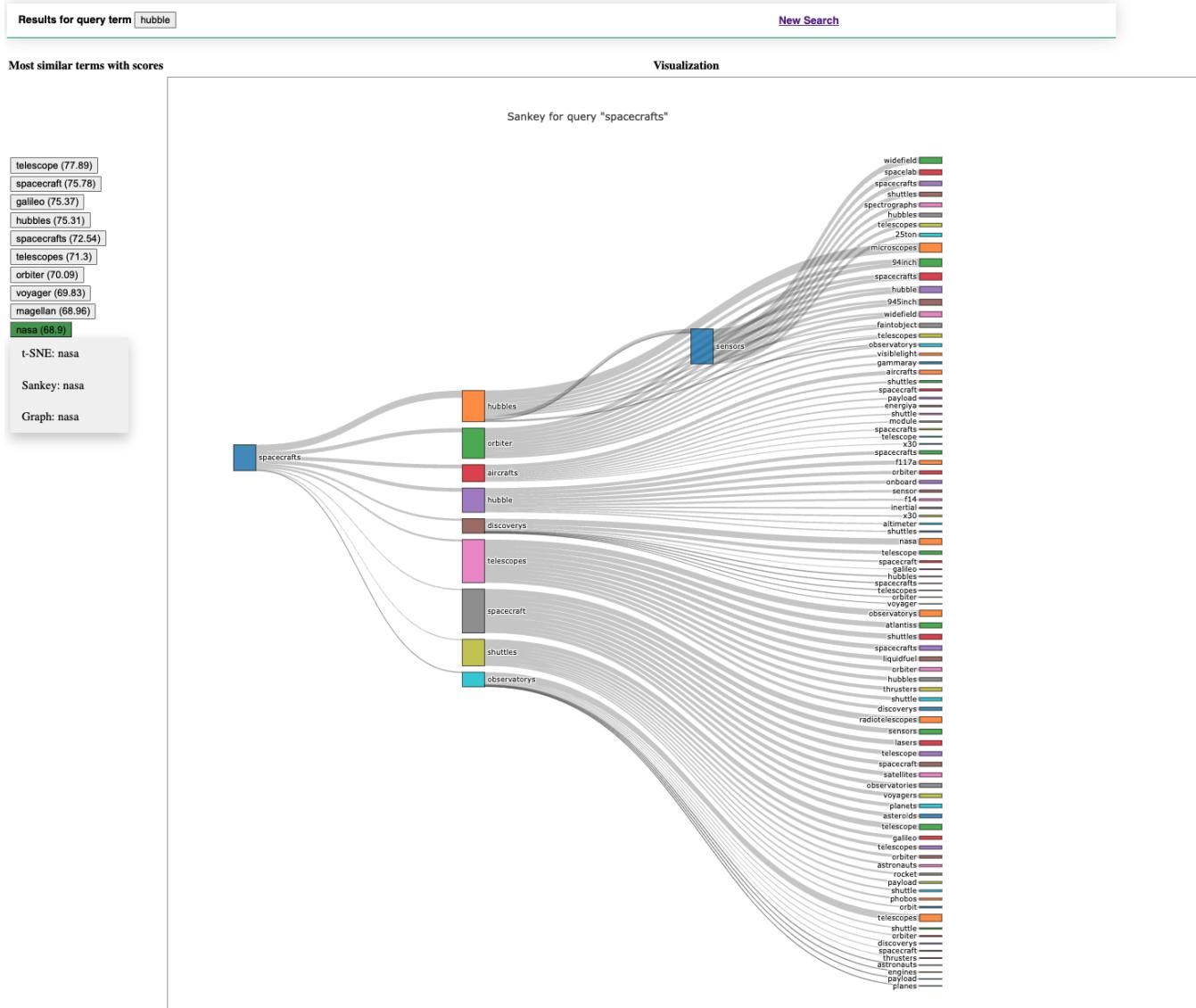
**Figure 3.** Search interface for an instance of the Word Embeddings Navigator exposing four temporally adjacent word embedding models for the New York Times annotated corpus (1987-1991, 1992-1996, 1997-2001, 2002-2006).

Embedding navigator supports several other methods for interactively querying an embedding model. The analogy search corresponds to a common means of evaluating word embedding models: X is to Y, as A is to B. The user provides three of the four elements of the analogy query. In its simplest form this sort of match would be performed using very basic vector arithmetic where the most similar left out term is the term closest to the vector  $Y-X+A$ . An important characteristic of the *gensim* library is that it includes implementations of many of the latest state of the art approaches to solving natural language processing tasks. For analogies, *gensim*’s *word2vec* class provides an improved method for the analogy task in the form of a method called *most\_similar\_cosmul*. This method uses a normalized multiplicative objective called 3COSMUL when calculating best matches for the left out analogy parameter. The exact method call looks like this:

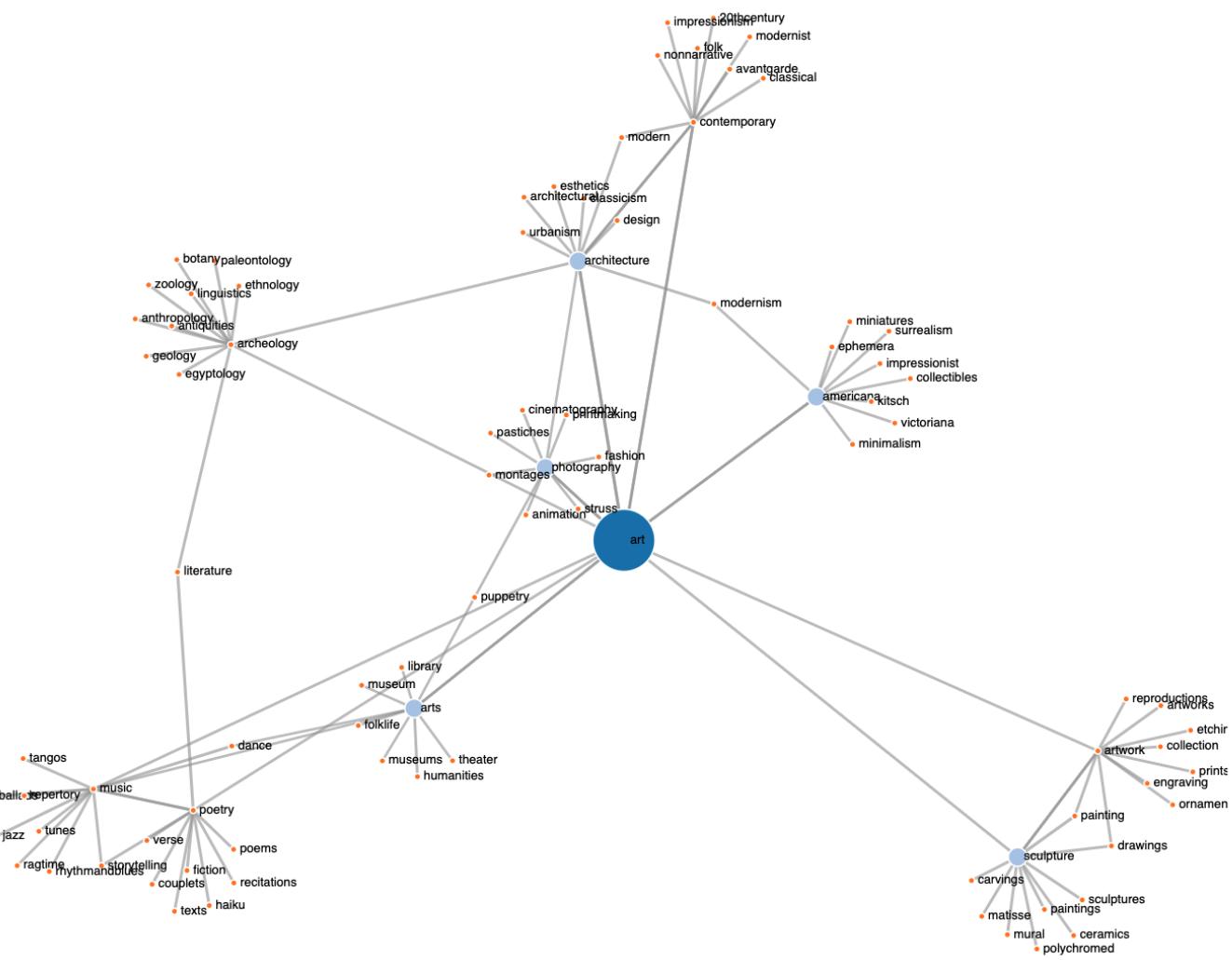
```
1 | this_model.w2v.most_similar_cosmul(
2 |     positive=[term1, term2], negative=[term3], topn=10
3 | )
```

Where X and A are the positive terms, Y is the negative term, and topn indicates the number of matches to return. When compared to simple vector math for solving analogy questions with word embeddings, **3COSMUL provides a 20% improvement** (Levy et al. 2014) in correctly identifying analogy relationships.

The *most\_similar\_cosmul* method is also used to identify terms that are most similar to a pair of terms. Although the actual calculation is a bit more complex, in essence, the best matches are those which are closest to the normalized average of the two query terms provided by the user. This is indicated by the inclusion of a positive parameter, and omission of a negative parameter passed to the method.

**Words close to "hubble" in nyt-87-91 embedding model****Figure 4.** A sankey diagram for the term “spacecrafts” from the 1987-1991 instance of the NYT corpus.

These four query options form the core of the embedding navigator search functionality. Results are available in several forms and the user can specify the number of results they wish to see at query time. The basic results form is a sorted list of most similar terms together with their respective cosine similarity scores. From this listing users can perform several tasks. The list provides click to query access for each term in the results set. There are also three visualization options provided: a t-SNE plot of the most similar terms, a sankey flow diagram of first and second order results, and a graph, or network visualization, which also presents first and second order results. Since two of the visualizations are graph based and it may not be immediately obvious how this would be achieved, we explain it here. First order results are those which are in close proximity to the original query term. Second order similar terms are the set of results associated with each first order search result. Thus the farthest nodes in the network visualization would generally be two degrees from the initial query term. The advantage of providing a network view is that it can reveal instances where a term has multiple connections among nodes in the first or second order result sets.



**Figure 5.** A force-directed graph of the term “art” together with its most similar terms in the embedding model.

**t-SNE** (Van der Maaten et al. 2008) is a technique for visualizing high dimensional data which preserves a reasonable degree of variation during dimensionality reduction. The true value of t-SNE is realized when visualizing both high dimensional data and a large number of data points. We use a javascript implementation of a t-SNE visualization tool which performs the dimensionality reduction in the browser. For small results sets, performance is adequate and we find that the plots it generates accurately reflect the relationships of the embedding vectors. While it is a tall order to expect good performance from browser-based dimensionality reduction of large amounts of high dimensional data, we include this capability because it provides a different perspective on a cluster of word embedding vectors. t-SNE remains a state of the art technique for visualizing high dimensional data.

As alluded to above, the other two visualizations provided are both based on network visualizations. The sankey diagram (Figure 4) shows the query term and two levels of results. The line connecting each result indicates how similar the term is to the term it is connected to. It is implemented using the javascript version of the [plotly library](#) [4], so it provides connection information when the user hovers over a result node, which is particularly useful when there is only a small amount of variation in the similarity scores. From the sankey diagram, the user can change the number of search results for the current query to see additional words in relation to the initial search term.

The user can also elect to view the result set as a graph (Figure 5). The graph implementation is a basic network diagram implemented in d3 [5]. The graph does not utilize similarity scores so edge width is uniform. This graph is intended to provide a comprehensive view of a term in relation to its nearest (most similar) neighbors in the embedding space. The graph uses a force directed layout, which is a layout model that tries to minimize overlap of nodes. The user can interact with nodes in the graph to further reduce overlap in a given region of the graph by selecting and dragging a node. A major advantage of the graph view is that recurring nodes are connected wherever they occur in the result set so that the user can see connections such as cycles and clusters that occur when the same term occurs in multiple results sets. These are relationships which are not depicted in the sankey view. So to summarize, the t-SNE view provides an approximate 2-D representation of term proximity, the sankey view links related terms and conveys similarity via node and edge width, while the graph view provides a more complex contextual perspective where first and second order results are visible.

## Adjusting word embeddings with interactive refitting

Human interaction with word embeddings allows for the injection of human judgment into the model. Humans are better at detecting bias and can use their expert knowledge to improve or correct other relationships in the embedding space. We are good at organizing terms according to more nuanced relationships, for example by emphasizing particular word groupings, flagging certain words as orthogonal to other words, reducing emphasis on syntactic relationships, and adjusting for the effects of bias. Interaction can occur after embeddings are generated via an unsupervised method. User provided modifications to the embedding space can be directly applied to word vectors or recorded and analyzed for consistency and resolution if re-fitting adjustments from different users target the same word or set of words.

There's an important caveat to the benefits of having a human identify and minimize bias: we all have implicit biases. Implicit biases are unconscious stereotypical associations we learn over the course of our life. No matter how hard we try, most of us bring implicit biases to situations and circumstances we encounter, even efforts to address bias. This includes interactively mitigating bias in word embeddings. If you lack awareness of your implicit biases, you might inadvertently substitute one kind of stereotype for another, thereby exacerbating bias rather than reducing it. There are online tests such as those provided by "Project Implicit" [6] and "Learning for Justice" [7], which allow individuals to assess their own implicit biases. Both tests are based on the "Implicit Association Test" (Bertrand et al, 2005) which is discussed in more detail below. It is good practice for individuals engaged in evaluating and mitigating bias in word embeddings to know their own biases beforehand. It is also recommended to incorporate input from multiple individuals when possible.

With that in mind, we can now look at how *WEN* enables users to interactively address bias in word embeddings. In their paper "[Retrofitting Word Vectors to Semantic Lexicons](#)" (Faruqui et al. 2014), the authors considered how to improve word embeddings by injecting additional semantic information into previously trained word embedding models. Their goal was to improve word embeddings by using relationships specified in a curated lexical dictionary such as [WordNet](#) [8] to adjust words that were identified as synonyms, so that their embedding vectors were more similar. They developed a light-weight post-processing approach that searches for embedding model terms in a lexical dictionary such as WordNet, identifies and locates the vectors of any synonyms, and moves the synonym vectors closer to the target term. Mathematically, their post-processing reduces the Euclidean distance of each synonym vector to the target term vector.

They were able to demonstrate that this approach improved representational semantics of the embedding vectors. Inspired by this work, we explored replacing data from a lexical dictionary with real-time human judgment. Instead of affecting the embedding model by moving synonyms, we implemented a mechanism whereby the user could interactively select words to be moved. Our project enables users to trigger one of two types of refitting of user-selected word vectors in the word embedding model using *WEN* (Figure 6). One moves a chosen term or list terms closer to a target word where the target word vector remains unchanged, while the other allows the user to indicate that the vectors for a set of terms should be moved closer to one another. We refer to this technique as [refitting](#) (Powell et al. 2020).

Refitting enables a *WEN* user to make adjustments to the embedding vectors of selected words and immediately see the results of these adjustments in subsequent queries and visualizations. Further adjustments can be made in an iterative fashion until the desired changes are achieved. Users can use whatever criteria they wish in order to make refitting adjustments.

Users are presented with two primary methods for modifying the embedding space: to move a single word closer to a list of related terms (Figure 7), or to move a set of words closer to one another (Figure 8). These affordances allow users to improve (decrease the distance) between embedding vectors. Each user interaction affects the vectors for the selected words. The recalculated word vectors are stored in a separate modified word embeddings table, along with a unique identifier, so that they can be used, analyzed, reclustered, and reconciled as needed. In addition to offering users the ability to refine relationships among words, users can take actions that compensate for bias in word embeddings, for example by moving gender specific pronouns closer to gender neutral pronouns.

**Perform Retrofit**

Mark as target	Add to semantic group	Term
<b>Select all <input checked="" type="checkbox"/></b>		
<input type="radio"/>	<input checked="" type="checkbox"/>	mary
<input type="radio"/>	<input checked="" type="checkbox"/>	elizabeth
<input type="radio"/>	<input checked="" type="checkbox"/>	theresa
<input type="radio"/>	<input checked="" type="checkbox"/>	catherine
<input type="radio"/>	<input checked="" type="checkbox"/>	anne
<input type="radio"/>	<input checked="" type="checkbox"/>	katherine
<input type="radio"/>	<input checked="" type="checkbox"/>	margaret
<input type="radio"/>	<input checked="" type="checkbox"/>	jane
<input type="radio"/>	<input checked="" type="checkbox"/>	stephanie
<input type="radio"/>	<input checked="" type="checkbox"/>	karen
<input type="radio"/>	<input checked="" type="checkbox"/>	carrie
<input checked="" type="checkbox"/>		Other term(s): <input type="text" value="science.scientist"/>

**Figure 6.** Example of the refitting interface which a user has configured to adjust a set of selected terms.

As noted above, the original retrofit objective is at the heart of the refitting strategy. We adapted the retrofit algorithm to support interactive word embedding adjustments and exposed it as a Web service used by *WEN*. Parameters include a set of words and optionally a target word. If a target is not provided, then each provided term will be adjusted using its original embedding vectors and the refitting objective so as to move it closer to all other terms provided. Otherwise, only the target embedding will be adjusted. The Web service returns the pre- and post-refitted vectors, as well as cosine distance scores among the terms that were refitted.

The target word embedding vectors are loaded into a dictionary from the word2vec model:

```
1 | for idx, key in enumerate(this_model.wv.vocab):
2 |     wordVecs[key] = this_model.wv[key]
```

The refitting function iterates over the entries in wordVecs from this\_model. The new embedding vector will be temporarily stored in castVector

```
1 | castVector = [float(i) for i in wordVecs[key]]
```

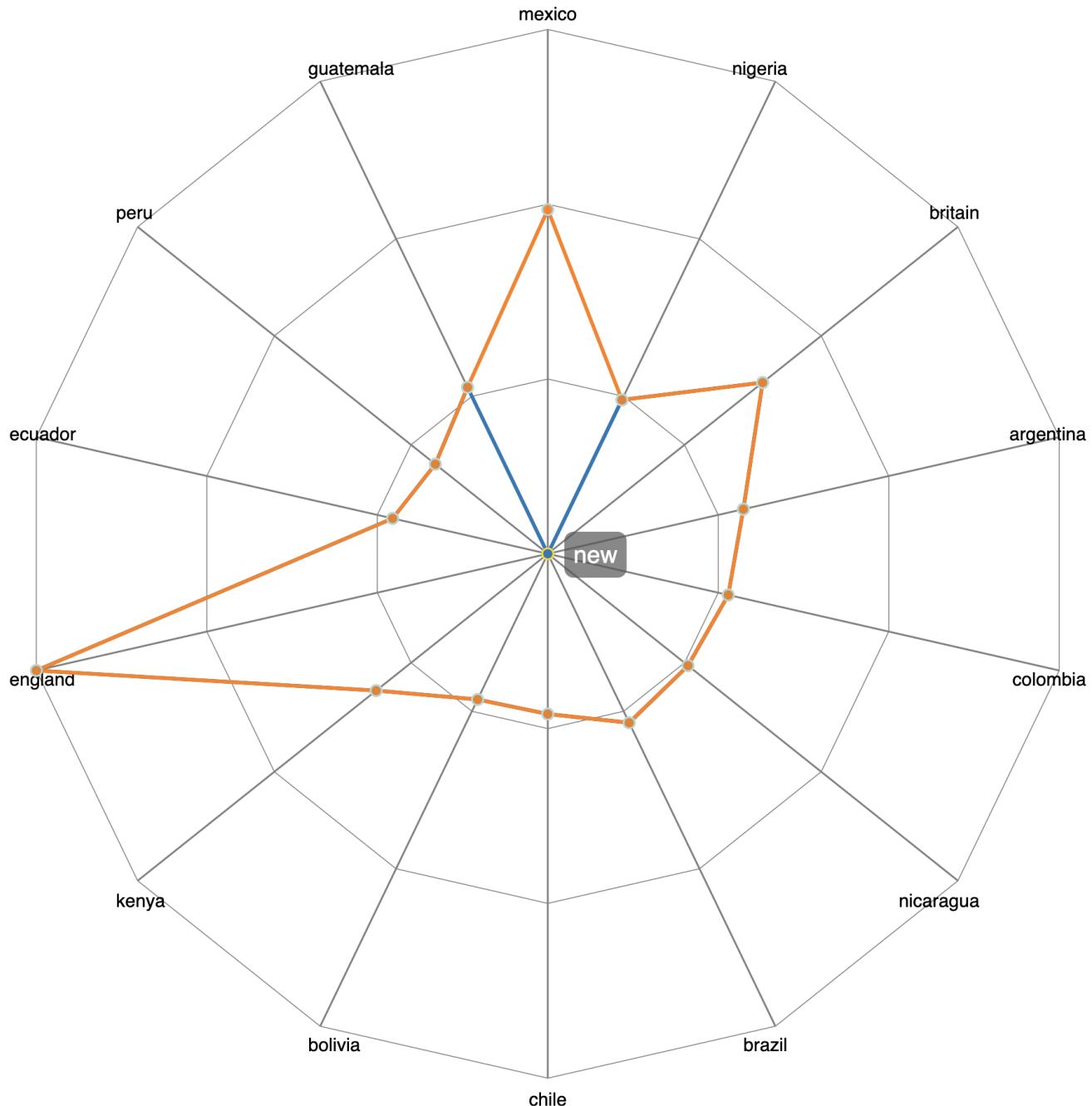
Once the word embedding vector is adjusted per user input, the refitted version is stored in the target word embedding model:

```
1 | this_model.wv.syn0[this_model.wv.vocab[key].index] = castVector
```

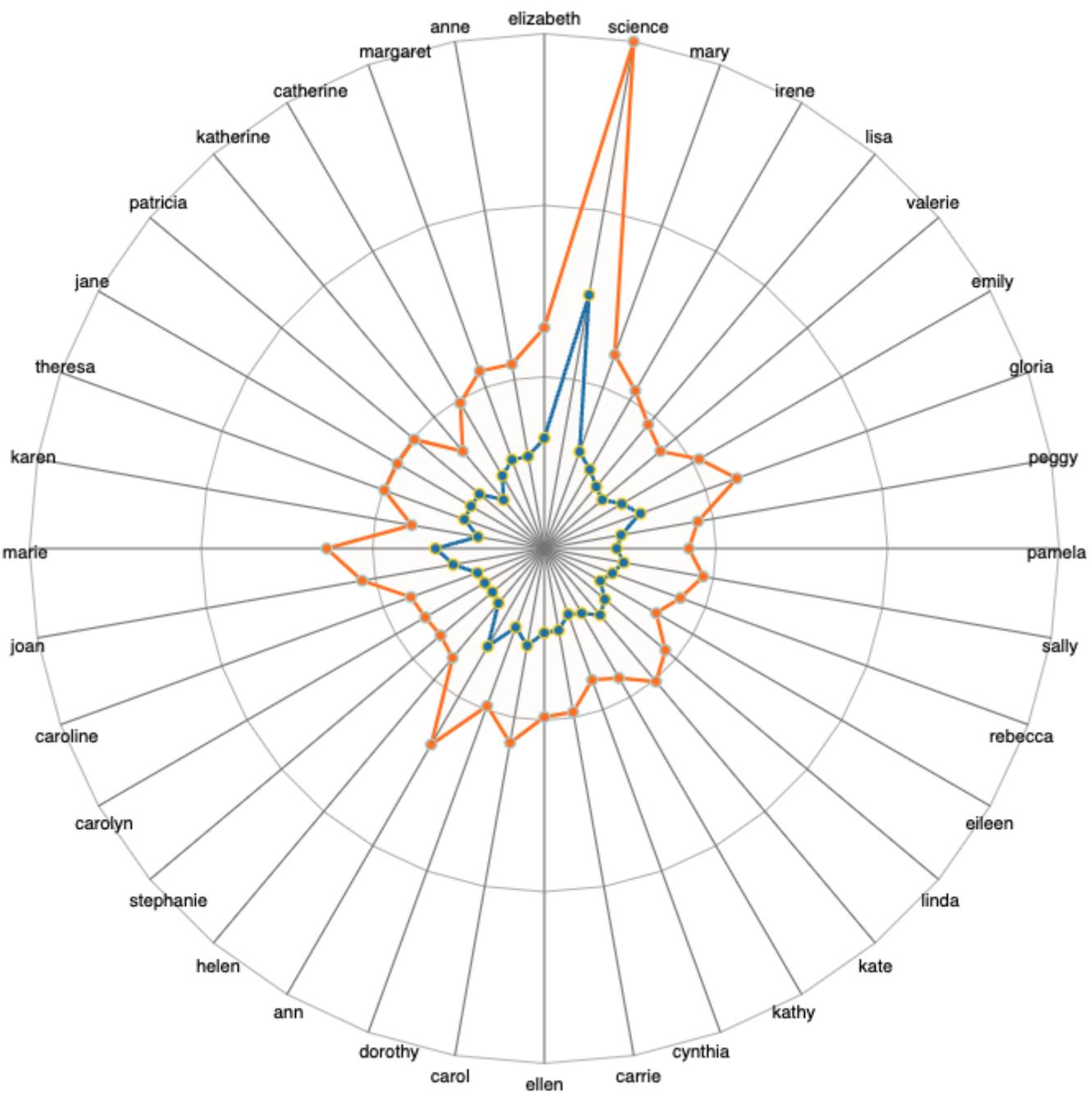
The old embedding vector for the word represented by the variable “key” is replaced with a new refitted embedding vector. This changes the in-memory version of the target word2vec model. To permanently serialize the updated version of the model to the filesystem, we call the save method of the target embedding model:

```
1 | this_model.save(model_filename)
```

When a refitting task is completed, a radar visualization is presented to show how the selected terms were moved in relation to one another, based on old and new cosine distances.



**Figure 7.** A before-and-after radar plot illustrating the effects of adjusting the word embedding vector for one term (“mexico”) so that it is closer to a list of other words (names of other countries).



**Figure 8.** A before-and-after radar plot illustrating the effects of refitting an entire set of terms.

Each time an embedding vector is adjusted, WEN stores the previous and updated vector in a MySQL database, together with the identifier for the target embedding model, an action label, and a timestamp. Although this is currently just a logging feature, the plan for this data in the future is to add an interface that would allow a user to “roll back” a selected vector to a previous state.

### Evaluating WEN refitting

We would be remiss if we were to discuss gender bias issues without acknowledging the various persistent historical biases including gender bias, that have been perpetuated in language for centuries, sometimes causing great harm. References to binary gender identities and to gender stereotypes are based on historical data and are referenced in order to provide clear examples for the purpose of potential bias mitigation techniques applied to data consumed by machine learning algorithms. It is not our intent to perpetuate or endorse any form of bias, or to make any claim that any historical social or cultural norms are superior to current norms. Furthermore, we note that the corpus and the examples used in our evaluation are all based on English language terms and publications, including news articles from the New York Times [9] spanning 1987-2006. Because of this, many of our examples may not be directly relevant to non-Western, non-English speaking audiences working with non-English corpora who are considering non-Western cultures, historical narratives, stereotypes, or biases.

Considerable effort has been invested in recent years in characterizing bias, especially gender bias, in word embeddings, for example (Brunet et al. 2019) and (Garg et al. 2018), and the impact on machine learning tasks such as machine translation (Savoli et al. 2021). From a technical perspective, two theories in

particular have dominated the investigation of how gender bias is represented in word embeddings: the role of word frequency and the potential existence of a gender-specific component in word embeddings. Word frequency has also been found to impede post-processing debiasing. We believe the value of a tool such as *WEN* is that it is agnostic with respect to the way in which bias was represented in a training corpus, or how it was propagated to a word embedding model. It likely would not scale as a production tool for mitigating bias in large embedding models but is intended to be an exploratory platform in which one may perform queries and adjustments in an iterative fashion, enabling users to perform experiments based on theory and intuition.

Our data set consists of word embeddings constructed from four five year snapshots of the annotated New York Times Annotated Corpus, which contains over 1.8 million articles from the New York Times spanning January 1987 until June 2007. We generated local word2vec word embeddings for each five year collection (1987-1991, 1992-1996, 1997-2001, and 2002-2006) after extracting the articles from the original XML source and performing some rudimentary text preprocessing which normalized all text to lowercase and removed most punctuation, except for punctuation which marked sentence boundaries. We used the gensim Word2Vec library with the cbow (continuous bag of words) model, eliminating any words that occurred fewer than four times in the corpus, trained for 20 epochs resulting in word vectors containing 100 dimensions per term. The sliding window for evaluating co-occurrences was left at its default value of 5, as were other parameters not explicitly listed here:

```
1 | model = Word2Vec(sentences=sentences, size=100, workers=4, min_count=4,
2 |           sample=0.05, sg = 0, iter=20, hs = 0)
```

Some details about the source corpora and resulting word embedding models, as well as some initial bias scores for the models, are provided in Table 1.

NYT article corpus year span	Word count	Embedding size	WEAT: career and family	WEAT: math and arts	WEAT: science and arts
1987-1991	490380	240723430	436398	1.48	0.84
1992-1996	391855	178927558	351059	1.64	0.95
1997-2001	448514	219950197	416824	1.68	1.1
2002-2006	447227	219969632	426894	1.68	1.04
					0.74

Table 1: Corpus and embedding models overview

Using *WEN* and refitting, we conducted several experiments to determine if we could in fact reduce gender bias as measured by the [Word-Embedding Association Test \(WEAT\)](#) (Caliskan et al. 2017). *WEAT* was inspired by the Implicit Association Test (*IAT*), which measured the response time of individuals who were asked to perform a word pairing test: pairing two words they found similar and two words they found different. *WEAT* uses the cosine distance between pairs of words “as analogous to reaction time in *IAT*” (Caliskan et al. 2017). Word pair lists for *WEAT* can be defined by the user, but standard sets of words are provided with the toolkit for measuring several types of bias. In the first two tests, a single adjustment was made to a set of terms and the resulting model was evaluated. In the third experiment, we combined the two strategies before measuring the effects. We applied identical strategies to all four models. The resulting *WEAT* scores were compared to the hard debiasing strategy described in (Bolukbasi et al. 2016).

WEAT scores				
Corpus	1. pronouns	2. personal names	A combination of 1 and 2	Hard debiasing using <i>debiaswe</i>
1987-1991	0.68	<b>0.46</b>	>0.49	0.48
1992-1996	0.94	0.89	0.76	<b>0.57</b>
1997-2001	0.86	0.79	0.71	<b>0.6</b>
2002-2006	0.65	<b>0.34</b>	0.51	0.5

Table 2: *WEN* refitting strategies versus application of hard debiasing. Best results are in bold.

The first two experiments were inspired by research into the causes of gender bias and the occurrence of gender stereotypes in word embeddings: the effect of [personal pronouns](#) on gender bias (Atir et al. 2018), and [the role played by the common usage of personal names](#) (Johns et al. 2019), specifically first names with respect to gender and science.

Experiment 1: the “personal pronouns” strategy is based on the notion that personal pronouns may play a significant role in gender bias that is learned by word embeddings. We perform a search for science, and then perform a refitting with “she, her, hers.” We performed the same task for each of the four embedding models and then computed the *WEAT* “science and art” bias score for each.

Experiment 2: the “personal names” strategy. This entailed first performing a search for female personal names in each embedding model. We searched for “mary” which is the most common female name from the last 100 years [per the US Social Security Administration \[10\]](#). We then selected all entries in this result set and refitted them with the terms “science” and “scientist” and again evaluated the results using *WEAT*.

In experiment 3 we combined strategies 1 and 2 in order on each embedding model, and then measured the results. Table 2 summarizes the results of the experiments.

For comparison, we used the [Word Embeddings Fairness Framework \(WEFE\)](#) [11], for debiasing and gender bias evaluation. The WEFE python library is to word embeddings and bias, what gensim is to NLP. It implements a variety of common debiasing strategies and metrics proposed in literature, thus collecting many state of the art approaches to these problems under a single umbrella. We selected WEFE’s hard debiasing strategy for comparison to the results of our refitting experiments.

Using it is straightforward: we first load each of the NYT embedding models as *gensim* KeyedVectors and then apply the hard debiasing strategy as illustrated by this code excerpt:

```
1 | debiaswe_wordsets = fetch_debiaswe()
2 |
3 | definitional_pairs = debiaswe_wordsets["definitional_pairs"]
4 | equalize_pairs = debiaswe_wordsets["equalize_pairs"]
5 | gender_specific = debiaswe_wordsets["gender_specific"]
6 |
7 | hd = HardDebias(verbose=False, criterion_name="gender").fit(
8 |   word2vec_model,
9 |   definitional_pairs=definitional_pairs,
10 |   equalize_pairs=equalize_pairs,
11 | )
12 |
```

```

13 gender_debiased_model = hd.transform(
14     word2vec_model, ignore=gender_specific, copy=True
15 )

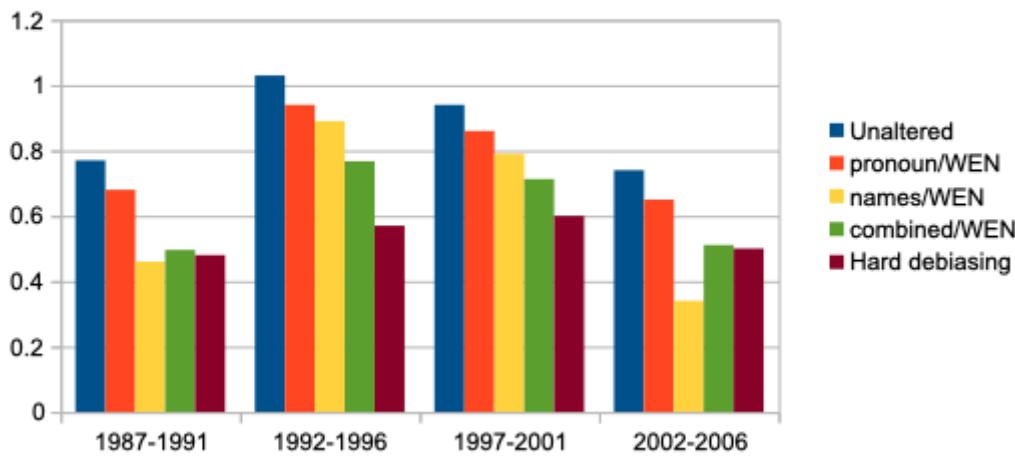
```

Three data sets are used by *debiaswe* hard debiasing to reduce gender bias. The definitional pairs data set includes pairs of words such as “woman” and “man” and “girl” and “boy.” The equalized pairs set is more role and career oriented, and includes pairs such as “king” and “queen” and “brother” and “sister.” The gender specific list appears to be learned from the original corpus and includes many additional social roles, biological gender specific topics such as “obstetrics” and “prostate cancer,” and many personal names. The first set is used to identify what the authors refer to as a “bias subspace.” This is used to determine how to neutralize bias. Words found in the gender specific list are omitted from bias neutralization. The remaining terms are adjusted by removing the bias direction identified from the bias subspace. Finally, equalized pairs are adjusted again so that they are equidistant from the vector representing the bias direction.

## Results and Conclusion

---

Gender bias mitigation



**Figure 9.** Plot which comparing the results of hard debiasing and several strategies applied using refitting in WEN on four temporal instances of the NYT corpus.

Prior to any debiasing efforts, Table 1 illustrates the gender bias scores for several areas of bias commonly found in text corpora: career and family, math and arts, and science and arts. The bias score from the *WEAT* metric can range from 2.0 to -2.0, where positive values indicate male gender bias, and negative values represent bias toward women. A score of zero suggests that gender bias does not exist per the test suite, or has been neutralized. We specifically focused on test cases related to “science” and “arts”, as the two topics seem distinct and queries for the two terms showed there was no overlap between them among their 200 nearest terms. Strategy 1 yielded only modest improvements, in the presence of gender bias as measured by the *WEAT* metric, moving the proverbially gender needle slightly away from male gender bias by between 8 and 12%. The personal names strategy was more successful, but varied dramatically from model to model, with a 13.5% shift for the 1992-1996 model, but a much more substantial 54% reduction in male gender bias for 2002-2006. Combining the two refitting strategies yielded consistently good results scoring which were consistently closer to the hard debiasing results than any other approach. Meanwhile, hard debiasing achieved better results for two models, and the improvements it made to the models were in general more consistent. Figure 9 illustrates the before and after impact of each strategy.

## Future work

---

There are several areas ripe for future exploration with *WEN*. A follow up paper inspired by the concept of retrofitting entitled “[Counter-fitting word vectors to linguistic constraints](#)” (Mrkšić et al. 2016) proposed a post-processing technique to introduce other information from lexical dictionaries. Notably this technique would identify antonym relationships and adjust word vectors accordingly. This could be readily adapted to *WEN*. It would be interesting to see how users would apply counterfeiting in a word embedding model. Would they focus strictly on antonym relationships or would they take advantage of the feature to reduce the proximity of terms for other reasons?

A full implementation of the capabilities enabled by refit logging would be another area of exploration. Questions such as how to select refitting actions for rollback, reversing a sequence of refitting tasks, and investigating the impact of selective rollbacks would be possible. Related to this would be expanding the somewhat neglected temporal capabilities of *WEN*. Would refitting across temporally aligned models (diachronic refitting) be a desirable feature?

More generally, *WEN* would benefit from additional methods of visualizing word embedding vectors. For example, the *plotly* python library supports a large number of visualizations and its performance is quite good. Related to this, it would be interesting to further explore what kinds of visualizations would best facilitate diachronic word analysis and latent knowledge discovery across temporally aligned word embeddings. Although exploration of temporal word embeddings inspired *WEN*, it lacks sufficient features to really explore words along a temporal dimension.

Finally, integrating bias metrics would be a highly desirable feature. But as with most debiasing-related approaches, these are designed to be run as post-processing tasks. Thus performance would likely be an issue, as would determining which metric(s) to implement. [Recent research raises questions about the effectiveness of WEAT in mitigating bias](#), for example. Some debiasing strategies fail to actually remove bias, even though they result in improved bias scores. This raises several questions, such as does *WEN* do better or worse at debiasing? Are there other bias metrics that more accurately measure bias? Are bias metrics suitable for integration with an interactive application? These are questions that could be explored in a future iteration of *WEN*.

*WEN* is primarily intended to be used with smaller, locally trained embedding models such as those based on the contents in an institutional repository, documents from a specific genre, or temporal embeddings where distinct models represent individual time slices of the original corpus. It allows users to test out different potential strategies to make adjustments to word embedding models, whether to neutralize bias, or to otherwise compensate for limitations in the source data upon which the model was trained. Locally trained models may have unique forms of bias that may not be effectively addressed by post-processing tools such as *WEFE*'s hard debiasing. We believe that *WEN* fills a niche in NLP / machine learning pipelines not adequately addressed by any existing tools.

## References

---

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Pennington, J., Socher, R. and Manning, C.D., 2014, October. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).
- Whitaker, B., Newman-Griffis, D., Haldar, A., Ferhatosmanoglu, H. and Fosler-Lussier, E., 2019. Characterizing the impact of geometric properties of word embeddings on task performance. *arXiv preprint arXiv:1904.04866*.
- Tsakalidis, A., Basile, P., Bazzi, M., Cucuringu, M. and McGillivray, B., 2021. DUKweb, diachronic word representations from the UK Web Archive corpus. *Scientific Data*, 8(1), pp.1-12.
- Bailey, A.H., Williams, A. and Cimpian, A., 2022. Based on billions of words on the internet, people= men. *Science Advances*, 8(13), p.eabm2463.
- Bolukbasi, T., Chang, K.W., Zou, J.Y., Saligrama, V. and Kalai, A.T., 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29.
- Brunet, M.E., Alkalay-Houlihan, C., Anderson, A. and Zemel, R., 2019, May. Understanding the origins of bias in word embeddings. In *International conference on machine learning* (pp. 803-811). PMLR.
- Savoldi, B., Gaido, M., Bentivogli, L., Negri, M. and Turchi, M., 2021. Gender bias in machine translation. *Transactions of the Association for Computational Linguistics*, 9, pp.845-874.
- Garg, N., Schiebinger, L., Jurafsky, D. and Zou, J., 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16), pp.E3635-E3644.
- Padilla, T., Allen, L., Frost, H., Potvin, S., Russey Roke, E. and Varner, S., 2019. Always already computational: collections as data. *Texas Digital Library* doi:10.5281/ZENODO.3152934
- Park, D., Kim, S., Lee, J., Choo, J., Diakopoulos, N. and Elmquist, N., 2017. Conceptvector: text visual analytics via interactive lexicon building using word embedding. *IEEE transactions on visualization and computer graphics*, 24(1), pp.361-370.
- Smilkov, D., Thorat, N., Nicholson, C., Reif, E., Viégas, F.B. and Wattenberg, M., 2016. Embedding projector: Interactive visualization and interpretation of embeddings. *arXiv preprint arXiv:1611.05469*.
- Katricheva, N., Yaskevich, A., Lisitsina, A., Zhordaniya, T., Kutuzov, A. and Kuzmenko, E., 2019, July. Vec2graph: A python library for visualizing word embeddings as graphs. In *International Conference on Analysis of Images, Social Networks and Texts* (pp. 190-198). Springer, Cham.
- Ghai, B., Hoque, M.N. and Mueller, K., 2021, May. WordBias: An Interactive Visual Tool for Discovering Intersectional Biases Encoded in Word Embeddings. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1-7).
- Sentz, K., Powell, J., Skurikhin, A., Porter, R. 2019. Searching for ConText: Microtasking to Solve Computationally Unsolvable Problems. LDRD Reserve Final Report LA-UR-19-30118.
- Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., Persson, K.A., Ceder, G. and Jain, A., 2019. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763), pp.95-98.
- Levy, O. and Goldberg, Y., 2014, June. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning* (pp. 171-180).
- Van der Maaten, L. and Hinton, G., 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).
- Bertrand, M., Chugh, D. and Mullainathan, S., 2005. Implicit discrimination. *American Economic Review*, 95(2), pp.94-98.
- Faruqui, M., Dodge, J., Jauhar, S.K., Dyer, C., Hovy, E. and Smith, N.A., 2014. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.
- Powell, J. and Sentz, K., 2020. Interactive Re-Fitting as a Technique for Improving Word Embeddings. *arXiv preprint arXiv:2010.00121*.
- Caliskan, A., Bryson, J.J. and Narayanan, A., 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334), pp.183-186.
- Greenwald, A.G., McGhee, D.E. and Schwartz, J.L., 1998. Measuring individual differences in implicit cognition: the implicit association test. *Journal of personality and social psychology*, 74(6), p.1464.
- Atir, S. and Ferguson, M.J., 2018. How gender determines the way we speak about professionals. *Proceedings of the National Academy of Sciences*, 115(28), pp.7278-7283.
- Johns, B.T. and Dye, M., 2019. Gender bias at scale: Evidence from the usage of personal names. *Behavior research methods*, 51(4), pp.1601-1618.
- Mrkšić, N., Séaghdha, D.O., Thomson, B., Gašić, M., Rojas-Barahona, L., Su, P.H., Vandyke, D., Wen, T.H. and Young, S., 2016. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*.
- Jo, H. and Choi, S.J., 2018. Extrofitting: Enriching word representation and its vector space with semantic lexicons. *arXiv preprint arXiv:1804.07946*.
- Gonen, H. and Goldberg, Y., 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *arXiv preprint arXiv:1903.03862*.

---

**End Notes**

- [1] Common Crawl. See <https://commoncrawl.org/>
- [2] "Teaching Method: Explanation" A discussion of explanation as a teaching method. See [https://en.wikipedia.org/wiki/Teaching\\_method#Explanation](https://en.wikipedia.org/wiki/Teaching_method#Explanation)
- [3] gensim "Topic Modeling for Humans" software library. See <https://radimrehurek.com/gensim/>
- [4] plot.ly visualization library. See <https://plotly.com/>
- [5] d3 Network graph gallery. See <https://d3-graph-gallery.com/network.html>
- [6] "Project Implicit." See <https://implicit.harvard.edu/implicit/takeatest.html>
- [7] "Test Yourself for Hidden Bias." See <https://www.learningforjustice.org/professional-development/test-yourself-for-hidden-bias>
- [8] WordNet. See <https://wordnet.princeton.edu/>
- [9] The New York Times annotated corpus. See <https://doi.org/10.35111/77ba-9x74>
- [10] "Top Names Over the Last 100 Years" from the US Social Security Administration. See <https://www.ssa.gov/oact/babynames/decades/century.html>
- [11] Documentation for "The Word Embeddings Fairness Evaluation Framework." See <https://wefe.readthedocs.io/en/latest/index.html>

---

**About the Authors**

James Powell ([jepowell@lanl.gov](mailto:jepowell@lanl.gov)) is a Research Engineer and a member of the Digital Library Research and Prototyping Team at the Research Library at Los Alamos National Laboratory, where he has worked for 17 years. Previously he worked at the University Libraries at Virginia Tech. His latest book is *A Librarian's Guide to Graphs, Data and the Semantic Web*. Chandos Information Professional Series. Waltham, MA: Chandos, 2015.

Kari Sentz ([ksentz@lanl.gov](mailto:ksentz@lanl.gov)) is currently a scientist in the Information Sciences Group (CCS-3) at Los Alamos National Laboratory. She has a Ph. D. in Systems Science from Binghamton University and a Master's in Linguistics from the University of Virginia. Sentz has worked at both Los Alamos and Sandia National Laboratories since 2000 researching generalized probability, text mining, risk analysis, probabilistic graphical modeling, and data and information fusion.

Elizabeth Moyer ([emoyer@lanl.gov](mailto:emoyer@lanl.gov)) is a librarian and member of the Reference and Research Services team at the Los Alamos National Laboratory Research Library. In this role, she helps support researchers by providing research support and is the Physics Liaison. She has an interest in the social bias of artificial intelligence through directed research conducted while she was a student at the University of British Columbia School of Information under the direction of Dr. Richard Arias-Hernández. She has continued interest and supports research in this domain for researchers at LANL through the development of a resource guide. Elizabeth holds a Masters of Library and Information Studies (MLIS) from the University of British Columbia School of Information in Vancouver, British Columbia, Canada.

Martin Klein ([mklein@lanl.gov](mailto:mklein@lanl.gov)), Los Alamos National Laboratory, is a scientist and lead of the Prototyping Team in LANL's Research Library. In this role, he focuses on research and development efforts in the realm of web archiving, scholarly communication, digital system interoperability, and data management. He is involved in standards and frameworks such as Memento, ResourceSync, Signposting, and Robust Links. Martin holds a Diploma in Computer Science from the University of Applied Sciences Berlin, Germany, and a Ph.D. in Computer Science from Old Dominion University.

Subscribe to comments: [For this article](#) | [For all articles](#)

---

This work is licensed under a Creative Commons Attribution 3.0 United States License.



See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/367828073>

# The Darkweb

Article · February 2023

---

CITATIONS

0

READS

114

1 author:



Shoumik Chandra

Lovely Professional University

1 PUBLICATION 0 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Research [View project](#)

# The Darkweb

Shoumik Chandra

School of Computer Science &

Engineering

Lovely Professional University

Phagwara, Punjab

e-mail:

chand rashoumik.1999@gmail.com

**Abstract**—*The Darkweb, is a hidden part of the internet that is not indexed by search engines and can only be accessed using specialized software, such as Tor. The Darkweb is often associated with illegal activities, such as the sale of drugs, weapons, and stolen data. However, it is also used for legitimate purposes, such as anonymity and free speech. This paper provides an overview of the Darkweb, including its definition, background, types of content, and the anonymity and security provided by it. Additionally, the paper discusses the impact of the Darkweb on society, efforts to regulate and shut down the Darkweb, current state and future trends, as well as ethical and moral issues associated with it. The paper concludes by emphasizing the importance of considering the potential harms and benefits of the Darkweb when discussing its regulation and use.*

**Keywords**—tor; hidden service; dark web; attack and defense of anonymous; dark web crawler; dark web data mining; understanding dark jargons

## I. INTRODUCTION

The dark web is a part of the internet that is not indexed by search engines and can only be accessed using specialized software, such as the TOR network. The anonymity provided by the dark web has made it a popular destination for illegal activities, such as the sale of illegal drugs and weapons, and the posting of stolen personal information. However, the dark web also has legitimate uses, such as for political dissidents and journalists operating in oppressive regimes, and for individuals seeking to protect their privacy. This review paper will provide an overview of the different components that make up the dark web, discuss the various methods used to attack anonymity on the dark web and the measures that can be taken to defend against such attacks, explore the use of dark web crawlers for data collection and the ethical and legal considerations of dark web data mining and understanding the dark web jargons.

## II. COMPONENT

The dark web is composed of several different components that work together to provide anonymity and access to hidden services. One of the most important components of the dark web is the TOR network.

TOR, short for The Onion Router, is a network of volunteer-run servers that allows users to browse the internet anonymously. The TOR network works by routing internet traffic through multiple layers of encryption, making it difficult to trace the origin of the traffic. The use of TOR provides a great deal of anonymity, but it also has some drawbacks. For example, the use of TOR can slow down internet speeds and make it more difficult to access some websites.

TOR is a free and open-source software that enables anonymous communication by routing internet traffic

through a network of volunteer-operated servers, known as nodes. Each node in the TOR network only knows the location of the previous and next node in the path of the internet traffic, making it difficult to trace the origin or destination of the traffic (Dingledine, Mathew, & Syverson, 2004).

Another important component of the dark web is hidden services. These are websites that can only be accessed through the TOR network, and their location and operators are concealed. Hidden services are created using the TOR network's hidden service protocol, which allows for the creation of virtual private networks (VPNs) on top of the TOR network (Murdoch & Danezis, 2007).

## III. TOR

TOR is a network of volunteer-run servers that allows users to browse the internet anonymously. The TOR network works by routing internet traffic through multiple layers of encryption, making it difficult to trace the origin of the traffic. The use of TOR provides a great deal of anonymity, but it also has some drawbacks. For example, the use of TOR can slow down internet speeds and make it more difficult to access some websites. Additionally, the anonymity provided by TOR is not foolproof, and there have been cases of users being de-anonymized through various methods such as traffic analysis and browser fingerprinting.

## IV. HIDDEN SERVICES

Hidden services are websites that can only be accessed through the TOR network and are not indexed by traditional search engines. These services are typically hosted on the dark web and provide a high level of anonymity for both the website owners and users. Examples of commonly used hidden services include drug marketplaces and whistleblower platforms.

One of the most well-known examples of a dark web hidden service is the Silk Road, which was a notorious online marketplace for illegal drugs. The Silk Road was eventually shut down by law enforcement, but similar marketplaces continue to exist on the dark web. Other examples of hidden services include platforms for anonymous communication and file sharing, such as the now-defunct platform known as Freedom Hosting.

The anonymity provided by the dark web has made it a popular destination for illegal activities, such as the sale of illegal drugs and weapons, and the posting of stolen personal information (Van der Meijden, & Van Eeten, 2018). The Federal Bureau of Investigation (FBI) estimates that the dark web is used for around 20% of all cybercrime (Cresci, 2015). The most popular marketplaces on the dark web are used to buy and sell illegal drugs, with the most popular being the now-defunct Silk Road (D'Angelo, 2016).

## V. LEGITIMATE USES OF THE DARK WEB

While the dark web is often associated with illegal activities, it also has legitimate uses. For example, political dissidents and journalists operating in oppressive regimes can use the dark web to communicate and share information without fear of censorship or surveillance (Kshetri, 2018). Additionally, individuals seeking to protect their privacy can use the dark web to communicate and share information without fear of being tracked or monitored.

## VI. DARK WEB

The dark web has a long and complex history, dating back to the early days of the internet. The first iteration of the dark web was a collection of underground bulletin board systems (BBS) that were only accessible via dial-up connections. As the internet evolved, so did the dark web. The development of TOR and other anonymity-providing technologies made it possible for individuals to access and operate hidden services on a larger scale.

## VII. METHODS OF ATTACKING ANONYMITY

There are several methods that can be used to attack the anonymity provided by the dark web. One method is traffic analysis, which involves analyzing patterns in internet traffic to identify the source and destination of the traffic (Panchenko, & Zajcev, 2016). Another method is browser fingerprinting, which involves identifying a user by their browser's unique configuration, such as installed fonts and browser plugins (Eckersley, 2010). Law enforcement agencies and other organizations have also developed methods for identifying and tracking users on the dark web, such as by infiltrating dark web marketplaces and using malware to track users (Europol, 2017).

## VIII. METHODS OF DEFENDING AGAINST ATTACKS

There are several measures that can be taken to defend against the methods of attacking anonymity on the dark web. One measure is using the TOR network's built-in security features, such as using the TOR browser and enabling the NoScript plugin (TOR Project, 2020). Another measure is using a VPN in conjunction with the TOR network to add an extra layer of encryption and protection (BestVPN, 2021). Additionally, using secure communication methods such as encrypted messaging apps can also help protect against attacks (WhatsApp, 2021; Signal, 2021). It is also important for users to be aware of the risks associated with using the dark web and to take steps to protect their identities.

## IX. DARK WEB CRAWLER

Dark web crawlers are specialized software programs that are used to automatically browse and collect data from hidden services on the dark web. These crawlers can be used for a variety of purposes, such as tracking illegal activities, identifying potential security threats, and conducting research.

However, the use of dark web crawlers raises several ethical and legal considerations. For example, the collection of data from hidden services without the consent of the operators or users may be considered a violation of privacy. Additionally, the use of dark web crawlers may also be illegal in certain jurisdictions.

## X. DARK WEB DATA MINING

Dark web data mining is the process of collecting, analyzing, and interpreting data from the dark web. This can include data from hidden services, forums, and other sources. The goal of dark web data mining is to gain insights into the activities and behaviors of users on the dark web.

Dark web data mining can be used for a variety of purposes, such as identifying potential security threats, tracking illegal activities, and conducting research. However, it also raises several ethical and legal considerations. For example, the collection of data from the dark web without the consent of the users may be considered a violation of privacy. Additionally, the analysis and interpretation of dark web data may also be subject to legal limitations.

## XI. UNDERSTANDING DARK WEB JARGONS

The dark web is often associated with illegal activities and is therefore associated with a lot of jargon and terminology. To understand the dark web, it is important to familiarize oneself with the jargons used on the dark web. For example, the term "onion" is often used as a metaphor for the layers of encryption and protection provided by the TOR network (TOR Project, 2020). Additionally, the term "clearnet" is used to refer to the regular internet, as opposed to the dark web (Kshetri, 2018). Other terms include "red rooms", which are urban legends of hidden services on the dark web where live torture and murder are streamed, but there is no evidence that these exist (Whitty, & Buchanan, 2019). Understanding these terms is important for anyone interested in the dark web, whether for research or personal use.

Some examples of common dark web jargon include:

- Cryptocurrency: The dark web primarily uses cryptocurrency for transactions. Bitcoin is the most widely used cryptocurrency on the dark web.
- Escrow: A system of trust used to facilitate transactions on the dark web. Escrow is used to ensure that both parties involved in a transaction are satisfied before releasing payment.
- PGP: PGP stands for "Pretty Good Privacy" and is a widely used encryption method on the dark web. It is used to secure communications and transactions.

## XII. IMPACT ON SOCIETY

The darkweb has had a significant impact on society, primarily because of the illegal activities that take place on it. The sale of illegal drugs on the darkweb has been linked to the opioid epidemic in the United States, and the sale of stolen credit card information has led to financial losses for individuals and businesses. Additionally, the anonymity provided by the darkweb has made it a breeding ground for hate speech and extremist ideologies.

## XIII. EFFORTS TO REGULATE AND SHUT DOWN THE DARKWEB

Law enforcement agencies around the world have made efforts to shut down darkweb markets and arrest the individuals behind them. In 2013, the FBI shut down Silk Road and arrested its operator, Ross Ulbricht. In 2017, the

Dutch police shut down the darkweb market AlphaBay. Furthermore, the US government passed the Allow States and Victims to Fight Online Sex Trafficking Act (FOSTA), which makes it a federal crime to operate a website that facilitates prostitution.

#### XIV. CURRENT STATE AND FUTURE TRENDS

Despite efforts to shut down darkweb markets, they continue to exist and new ones are constantly being created. The darkweb is also becoming increasingly decentralized, with decentralized marketplaces and peer-to-peer networks making it more difficult for law enforcement to shut them down. Additionally, the anonymity provided by the darkweb is being used for legitimate purposes such as whistleblowing and political activism.

#### XV. ETHICAL AND MORAL ISSUES

The use of the darkweb raises several ethical and moral issues. The anonymity provided by the darkweb can be used to protect human rights and freedom of speech, but it also allows for the sale of illegal drugs and weapons, which can harm individuals and society as a whole. Additionally, the use of the darkweb for illegal activities raises questions about personal responsibility and accountability.

#### XVI. CONCLUSION

The dark web is a complex and multifaceted ecosystem, which has both legitimate and illegitimate uses. Understanding the different components of the dark web, as well as the methods used to attack and defend anonymity on the dark web, is crucial for both individuals and organizations. Additionally, the use of dark web crawlers and data mining on the dark web must be done with consideration for ethical and legal issues. And, to understand the dark web, it's important to be familiar with the jargons and legends used on the dark web.

Darkweb is a network of hidden websites and services that can only be accessed using specialized software, such as Tor. It is primarily used for illegal activities such as buying and selling drugs, firearms, and stolen credit card information. The anonymity provided by the darkweb can be used for both good and bad purposes, and its use raises several ethical and moral issues.

#### ACKNOWLEDGMENT

I would like to express my gratitude to the researchers and experts in the field of darkweb studies

whose work has served as the foundation for this review paper. Finally, I would like to acknowledge the invaluable input and feedback provided by the reviewers, whose suggestions have greatly improved the quality and accuracy of this paper.

#### REFERENCES

- BestVPN. (2021). The Best VPNs for TOR: Protect Your Privacy on the Dark Web. BestVPN.
- Cresci, S. (2015). How the DarkWeb is used for cybercrime. The Guardian.
- D'Angelo, J. (2016). Inside the DarkWeb: A Tour of the Underground Internet. Rolling Stone.
- Dingledine, R., Mathewson, N., & Syverson, P. (2004). Tor: The second-generation onion router. In Proceedings of the 13th USENIX Security Symposium (pp. 303-320).
- Eckersley, P. (2010). How unique is your web browser? Electronic Frontier Foundation.
- Europol. (2017). Internet Organised Crime Threat Assessment (IOCTA) 2017. Europol.
- Kshetri, N. (2018). The dark side of the Internet: the criminal exploitation of the dark web. Journal of Cybersecurity, 4(2), 83-99.
- Murdoch, S. J., & Danezis, G. (2007).
- "Darknet Markets." Europol, European Union Agency for Law Enforcement Cooperation, 2020.
- "Silk Road." The New York Times, The New York Times.
- "Shutting Down AlphaBay, the World's Largest Criminal Marketplace on the Darknet." Europol, European Union Agency for Law Enforcement Cooperation, 2017.
- "Allow States and Victims to Fight Online Sex Trafficking Act of 2017." [Congress.gov](#), Library of Congress.
- "Darknet and Bitcoin." In Bitcoin and Cryptocurrency Technologies, edited by Arvind Narayanan et al., Princeton University Press.
- "On the Underground Economy and Its Relationship to the Cybercrime Ecosystem." In Cybercrime and Espionage: An Analysis of Subversive Multi-Vector Threats, edited by Sean E. Sawyer et al., Springer, 2015.
- "The Ethics of Anonymity." In The Oxford Handbook of Information Ethics, edited by Luciano Floridi, Oxford University Press, 2011.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/233501081>

# Using corpora in machine-learning chatbot systems

Article in International Journal of Corpus Linguistics · December 2005

DOI: 10.1075/ijcl.10.4.06sha

---

CITATIONS  
149

READS  
8,043

---

2 authors:



Bayan Abu Shawar  
Al Ain University  
60 PUBLICATIONS 1,274 CITATIONS

[SEE PROFILE](#)



Eric Atwell  
University of Leeds  
408 PUBLICATIONS 4,445 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



SnoMedTagger: A semantic tagger for medical narratives using SNOMED CT [View project](#)



SCUoL at CheckThat! 2022: fake news detection using transformer-based models [View project](#)

# Using corpora in machine-learning chatbot systems

Bayan Abu Shawar and Eric Atwell  
University of Leeds

A chatbot is a machine conversation system which interacts with human users via natural conversational language. Software to machine-learn conversational patterns from a transcribed dialogue corpus has been used to generate a range of chatbots speaking various languages and sublanguages including varieties of English, as well as French, Arabic and Afrikaans. This paper presents a program to learn from spoken transcripts of the Dialogue Diversity Corpus of English, the Minnesota French Corpus, the Corpus of Spoken Afrikaans, the Qur'an Arabic-English parallel corpus, and the British National Corpus of English; we discuss the problems which arose during learning and testing. Two main goals were achieved from the automation process. One was the ability to generate different versions of the chatbot in different languages, bringing chatbot technology to languages with few if any NLP resources: the corpus-based learning techniques transferred straightforwardly to develop chatbots for Afrikaans and Qur'anic Arabic. The second achievement was the ability to learn a very large number of categories within a short time, saving effort and errors in doing such work manually: we generated more than one million AIML categories or conversation-rules from the BNC corpus, 20 times the size of existing AIML rule-sets, and probably the biggest AI Knowledge-Base ever.

**Keywords:** chatbot, dialogue, AIML, Artificial Intelligence, Machine Learning, French, Afrikaans, Arabic, Qur'an, British National Corpus, lemmatised and unlemmatised lists.

## 1. Introduction

Corpora have been widely used by linguists to develop and refine “language models”, descriptions of lexis, grammar, dialogue, etc. Language models can

also be automatically extracted or machine-learnt from corpora, to drive language analysis systems; for example, machine-learning of Part-of-Speech taggers from PoS-tagged corpora (Atwell 1983; Atwell et al. 2000a); machine-learning to automatically cluster words in a corpus into grammatical classes (Atwell & Drakos 1987; Hughes & Atwell 1994); machine-learnt grammar checkers (Atwell 1983, 1987); machine learning of parsers from parsed corpus treebanks (Atwell 1983, 1988, 1996; Atwell et al. 1984), adapting speech recognition models for English language learners (Atwell et al. 2000b, 2003).

Chatbot technology integrates a language model and computational algorithms to emulate informal chat communication between a human user and a computer using natural language. The idea of chatbot systems originated in the Massachusetts Institute of Technology (Weizenbaum 1966, 1967), where Weizenbaum implemented the ELIZA chatbot to emulate a psychotherapist. The idea was simple and based on keyword matching. The input is inspected for the presence of a keyword. If such a word is found, the sentence is mapped according to a rule associated with the keyword; if not, a connected free remark, or under certain conditions an earlier transformation, is retrieved. For example, if the input includes the keyword “mother”, ELIZA can respond “Tell me more about your family”. This rule is inspired by the theory that mother and family are central to psychological problems, so a therapist should encourage the patient to open up about their family; but the ELIZA program does not really ‘understand’ this psychological strategy, it merely matches the keyword and regurgitates a standard response. To keep the conversation going, ELIZA has to produce responses which encourage the patient to reflect and introspect, and this is done mechanistically using some fixed phrases if no keyword match is found such as “Very interesting. Please go on.”, or “Can you think of a special example?”. The next major program was PARRY (Colby 1973, 1999). In contrast to ELIZA, instead of simulating a psychotherapist, PARRY modelled a paranoid patient during an interview with his therapist. Saygin, Cicekli and Akman (2000) noticed that, “Both ELIZA and PARRY use certain tricks to be able to successfully perform in conversations. ELIZA directs the conversation away from herself by asking questions. ELIZA uses parts of the user’s input in the output questions and seems to be following the conversations. In addition to these techniques, PARRY has little stories to tell and tend to insert these in the conversation.” Chatting (1) shows a sample of chatting with ELIZA.

---

**Chatting (1). A sample of conversation with ELIZA (from Jurafsky & Martin 2000)**

Human: Men are all alike.  
ELIZA: In what way?  
Human: They're always bugging us about something.  
ELIZA: Can you think of a special example?  
Human: Well, my boyfriend made me come here.  
ELIZA: Your boyfriend made you come here?  
Human: He says I'm depressed much of the time.  
ELIZA: I'm sorry to hear that you are depressed much of the time.  
Human: It's true. I'm unhappy.  
ELIZA: Do you think coming here will help you not to be unhappy?  
Human: I need some help, that much seems certain.

---

Since that time, several other chatbots or human-computer dialogue systems have been developed using text communication such as MegaHAL (Hutchens 1996), CONVERSE (Batacharia et al. 1999), ELIZABETH (Abu Shawar & Atwell 2002), HEXBOT (HEXBOT 2004) and ALICE (ALICE 2002). Chatbots have been used in different domains such as: customer service, education, web site help, and for fun.

However, these chatbots are restricted to the linguistic knowledge that is manually “hand-coded” in their files. To save the time and effort of encoding such knowledge and to develop a chatbot that simulates a human dialogue, we developed a Java program to convert a dialogue transcript text corpus to AIML format: Artificial Intelligence Markup Language, the ALICE chatbot rule-format (see Section 2). In order to retrain ALICE, we used a range of corpora to create several different experimental versions of ALICE, speaking different varieties of English, as well as French, Afrikaans, Arabic and bilingual chatbots. This paper illustrates the ability of our program to learn a linguistic knowledge base of more than one million categories or rules, extracted from the British National Corpus (BNC) spoken transcriptions. The approach seemed straightforward at the outset, but we encountered problems and drawbacks; we discuss these and propose potential directions for further research.

The ALICE chatbot engine and its AIML knowledge representation formalism are presented in Section 2. Section 3 outlines our initial attempts to learn AIML files from English, French, Afrikaans and Arabic corpora; we explain how feedback from users of our initial machine-learnt chatbots led us to

develop more sophisticated versions of the learning algorithm. Section 4 examines the British National Corpus and the problems which arose when converting the BNC spoken transcripts to the AIML format. The latest version of the AIML-learning program tackles the BNC problems; the necessary modifications are discussed in Section 5. The results and conclusions are in Sections 6 and 7 respectively.

## 2. The ALICE chatbot engine

A.L.I.C.E. (ALICE 2002; Wallace 2003) is the Artificial Linguistic Internet Computer Entity, first implemented by Wallace in 1995. ALICE knowledge about English conversation patterns is stored in AIML files. AIML, or Artificial Intelligence Mark-up Language, is a derivative of Extensible Mark-up Language (XML), developed by Wallace and the Alicebot free software community during 1995–2000 to enable people to input dialogue pattern knowledge into chatbots based on the ALICE open-source software technology.

AIML consists of data objects called AIML objects, which are made up of units called topics and categories. The topic is an optional top-level element, has a name attribute and a set of categories related to that topic. Categories are the basic units of knowledge in AIML. Each category is a rule for matching input to output, and consists of a pattern, which matches against the user input, and a template, which is used in generating the ALICE chatbot answer.

The AIML pattern is simple, consisting only of words, spaces, and the wildcard symbols \_ and \*. The words may consist of letters and numerals, but no other characters, as shown in Section 4.1.4. Words are separated by a single space, and the wildcard characters function like words. The pattern language is case invariant. The idea of the pattern matching technique is based on finding the best, longest, pattern match.

### 2.1 Types of ALICE/AIML Categories

There are three types of categories: atomic categories, default categories, and recursive categories.

- a. *Atomic categories* have patterns that do not have wildcard symbols \_ or \*, e.g.:

```
<category><pattern>Hello Alice</pattern>
<template>Hi, who are you?</template></category>
```

In the above category, if the user inputs ‘Hello Alice’, then ALICE answers ‘Hi, who are you?’. An atomic category only fires if the human input is an exact word-for-word match for the pattern; this can be used to encode formulaic conversation openers, for example.

- b. *Default categories* have patterns including wildcard symbols \* or \_. The wildcard symbols match any input but they differ in their alphabetical order. Assuming the input ‘Hello robot’, if this does not match a category with an atomic pattern, then it will try to find a category with a default pattern such as:

```
<category><pattern>Hello *</pattern>
<template>Hi there</template> </category>
```

So ALICE answers ‘Hi there’. The wildcard symbol allows the category to match a wider range of possible human inputs.

- c. *Recursive categories* have templates including <srai> and <sr> tags, which refer to *simply recursive artificial intelligence* and *symbolic reduction*. Recursive categories have many applications: symbolic reduction that reduces complex grammatical forms to simpler ones; divide and conquer that splits an input into two or more subparts, and combines the responses to each; and dealing with synonyms and misspellings by mapping different ways of saying the same thing to the same reply.

### c.1 *Symbolic reduction*

```
<category> <pattern>DO YOU KNOW WHAT THE * IS</pattern>
<template><srai>What is <star/></srai></template></category>
```

In this example <srai> is used to reduce the human input “Do you know what the \* is?” to a simpler form “what is \*”; this is then recursively fed back into ALICE as replacement for the original input, allowing other categories to match.

### c.2 *Divide and conquer*

```
<category><pattern>YES *</pattern>
<template><srai>YES</srai><sr/><template></category>
```

The input is partitioned into two parts, ‘yes’ and the second part; \* is matched with the <sr/> tag (where <sr/> is a short form for <srai><star/></srai>). Each of the two parts is then recursively fed back into ALICE to match against

separate categories; then the two separate responses are joined together to produce the ALICE reply.

### c.3 *Synonyms and misspellings*

```
<category><pattern>HALO</pattern>
<template><srai>Hello</srai></template></category>
```

The misspelled input ‘halo’ is mapped to another form ‘hello’, which has the same meaning and should match against another category when recursively fed back into ALICE.

## 2.2 ALICE Interpreter

The AIML interpreter tries to find the longest match between the human user input and the stored knowledge. The Interpreter does not have a record of the dialogue so far, so each ALICE response is based solely on the most recent human input. Furthermore, ALICE has no model of the overall goals of the dialogue, or what the user is “on about” beyond the most recent input. This may seem implausibly simple to a linguist; but chatbot developers have found that in practice a lot of conversations have little discourse structure, this “naïve” approach actually works much of the time as long as ALICE has a wide range of categories to respond to a wide range of possible human inputs. There are more than 50,000 categories in the current public-domain ALICE “brain”, added by the Botmaster over several years. The AIML format was deliberately kept simple to make it easy to author new rules or categories; other public-domain chatbots such as Elizabeth (Abu Shawar & Atwell 2003a, c) or HEXBOT (HEXBOT 2004) claim to be readily adaptable and extendable, but their more complex rule format makes it much harder to author extensions or new chatbot rules. However, all these categories are manually “hand-coded”, which even in AIML is time-consuming, and restricts adaptation to new discourse-domains and new languages. Chatting (2) shows a sample of chatting with the hand coded version of ALICE. In the following sections we will present the automation process we developed, to train ALICE using a corpus based approach.

**Chatting (2). A sample of conversation with ALICE (from the AI movie website)**

---

Emily: My name is Emily  
Chatbot: OK, I will call you Emily  
Emily: Are you really Alice?  
Chatbot: I won't say "yes" or "no" for now. For real  
Emily: Who created you?  
Chatbot: I was created by Dr Richard S. Wallace  
Emily: Do you know the chatbot Alice?  
Chatbot: Is there only one chatbot. Ask the open directory about it  
Emily: Have you seen the movie AI?  
Chatbot: Have you told many people that your favorite movie is a you?  
Emily: Can you explain?  
Chatbot: There is no more information to clarify.  
Emily: Do you know Steven Spielberg?  
Chatbot: Not a robot. Probably a human. Maybe a celebrity. Would you like to know more?  
Emily: I know. I was wondering if you knew.  
Chatbot: How do you know it? I have never considered it. I answer a lot of silly questions.  
Emily: You are actually a guest on the AI website. Do you know that?  
Chatbot: Thanks for telling me that I am actually a guest on the AI website  
Emily: What do you like to talk about?  
Chatbot: My favorite subject is artificial intelligence and robots.  
Emily: What is artificial intelligence?  
Chatbot: Artificial intelligence is the branch of engineering and science devoted to constructing machines that think. Would you like to know more?

---

### 3. Learning to chat in a range of languages

We developed Java software to extract AIML categories from a corpus text-file. Different corpora were used (displayed in Table 1) and a range of increasingly sophisticated learnt patterns were examined. The learning techniques range from primitive literal matches to corpus utterances, generating atomic categories which require an exact match to human user input, to more complicated patterns involving identification of the most significant words in an utterance, generating default categories with wildcards (Abu Shawar & Atwell 2003c).

**Table 1.** The corpora used to train our prototype chatbot-learning program

Corpus	Number of words	Size (KB)
Minnesota French Corpus	25,761	428
Corpus of Spoken Afrikaans	30,793	317
Qur'an in Arabic	85,229	779
Qur'an in English	175,626	955

### 3.1 Learning from the Dialog Diversity Corpus of English

The first version learnt simple pattern+template categories, where each utterance or turn in the dialogue was taken as a pattern to match the user input, and the subsequent or following utterance became the template for the chatbot answer. The program is composed of four phases: reading the dialogue from the corpus, and inserting it in a vector; applying a text-reprocessing module to remove all unnecessary annotations; passing over the converter module, which considers each turn as a pattern and its successor as a template; And finally saving these categories in an AIML file. This version was tested using samples of the English-language Dialogue Diversity Corpus (DDC) (Mann 2002). The DDC is a collection of links to different dialogue corpora in different fields where each corpus has its own annotation format. These annotated texts are transcribed from recorded dialogues between more than two speakers. Abu Shawar and Atwell (2003a) detail problems encountered, summarised as follows:

- a. No standard formats to distinguish between speakers, or for linguistic annotations.
- b. Extra-linguistic annotations were used.
- c. Long turns and monologues.
- d. Irregular turn taking (overlapping).
- e. More than two speakers.
- f. Scanned text-image not converted to text format.

Unfortunately most of these problems also occur in other corpora, which necessitates changing the filtering process to meet the difference in the corpora format. Figure 1 shows samples of the DDC corpora. The figure illustrates some of the above problems: speaker turns are marked “S1:”, “S2:” etc in MICASE, but by more complex XML tags in ICE; MICASE uses XML-like tags for extralinguistic annotations like “<SS LAUGH>” or “<ROTATES CEILING>”, these tags must be ignored.

---

*MICASE corpus:* Michigan Corpus of Academic Spoken English (Mann 2002) is a 1.8-million-word collection of transcripts of academic speech events recorded at the University of Michigan.

S1: circum polar stars. So if I keep my pointer there, [S2: oh ] <ROTATES CEILING> ev-  
erything else moves and we all get sick. <SS LAUGH> and we go backwards in time.  
And that's even more fun.

S2: make it go really really fast.

<SS LAUGH>

S1: okay so that's how the sky is going to move, a couple of other things that we can do in  
here, um, this is a presentation of, the, grid, that we use to divide the sky, so these lines  
that run, north south what do we call those?

S3: declination

*ICE-Singapore:* International Corpus of English, Singapore English (Nelson 2002), has one  
million words.

<\$A>  
<ICE-SIN:S1A-099#35:1:A>  
Uhm okay lah  
<ICE-SIN:S1A-099#36:1:A>  
Bearing up lah  
<\$B>  
<ICE-SIN:S1A-099#37:1:B>  
Ah hah  
<\$A>  
<ICE-SIN:S1A-099#38:1:A>

Ya I mean I don't really feel comfortable talking about it over the phone so when I see you  
I'll tell you about it lah

---

**Figure 1.** Samples of MICASE and ICE-Singapore subcorpora in the Dialog Diversity  
Corpus

### 3.2 Learning from the Minnesota French Dialogue Corpus

A great attraction of the Machine Learning approach is that a learning system used on English corpora should be readily applicable to corpora in other languages. The chatbot-training mechanism does not “understand” the dialogue, it simply treats it as a sequence of character-string-matches, and the character-strings could be in any language. To test this, we applied the same program to the Minnesota French dialogue corpus (Kerr 1983); this required chang-

---

MINNESOTA CORPUS  
SESSION I, TAPE 1, SIDE A

Christine=C.; Martine=M.; Evelyne=E.; unknown speaker=u.s.

- C. Peut-être il faut, un divan oui
- M. Vous allez [faire quoi] ce dimanche  
u.s. [ /inaudible / ]
- M. J'en avais un à dîner [l'autre fois euh]
- E. [Un divan?] Pourquoi tu vas pas euh, euh tu sais près de, tu connais Como, la, Como Avenue? Bon et là il y a c'est c'est l'Armée du Salut quelque chose et des fois y a de très très [jolies choses]  
C. [ah mm]
- E. et ça par exemple ça dépend ce que tu aimes mais il avait un divan imitation ancien pas an- ancien tu vois avec du bois, le devant ici en bois, et ici vraiment bien et c'est pas cher. Tu devrais de temps en temps y aller. Parce que nous on aller regarder aussi, bon enfin maintenant on a tout ce qui faut mais, euh c'est pas cher du tout et puis c'est pas c'est pas
- C. [mm mm]
- 

Figure 2. Sample of the Minnesota French corpus

ing the pre-processing text since it has its own specific annotations, illustrated in Figure 2. This figure shows that speaker turns are marked differently from MICASE and ICE: a single letter abbreviation of the speaker's name. The figure also shows overlapping is encoded via position of the text: for example, as Martine finishes her second utterance "... l'autre fois euh", Evelyn interrupts with "Un divan? ...". We were able to call on a number of French speakers in our Computer Vision and Language Laboratory to test the French chatbot.

### 3.3 Learning from the Corpus of Spoken Afrikaans

Our Machine Learning approach should be usable on languages with little or no existing chatbots or other language processing technology; and on languages which we do not speak ourselves, or have ready access to native-speaker informants. The only requirement is a corpus of spoken dialogue in the language in question. Gerhardt van Huysteen and Bertus van Rooy of Potschefstroom University suggested Afrikaans as a suitable language for our next trials, as they were able to give us access to the recently-collected Corpus of Spoken Afrikaans (Van Rooy 2003).

Our revised version of the learning program, Afrikaans.java, has a more general approach to finding the best match against user input from the training dialogue. Two machine learning techniques were adapted, the “first word” approach, and the “most significant word” approach.

In the first word approach we assumed that the first word of an utterance may be a good clue to an appropriate response: if we cannot match the input against a complete corpus utterance, then at least we can try matching just the first word of a corpus utterance. For each atomic pattern, we generated a default version that holds the first word followed by wildcard to match any text, and then associated it with the same atomic template.

The first word approach was tested using the Corpus of Spoken Afrikaans, illustrated in Figure 3. Speaker turns are encoded in yet another way: turns start with an XML “<sprekerN>” tag, and must also end with a matching “</sprekerN>” closing tag. Overlaps are also encoded via XML tags: <oovleuel> and closing tag “</oovleuel>”. Unfortunately this first word approach still failed to satisfy our trial users, so we looked for the word in the utterance with the highest “information content”, the word that is most specific to this utterance compared to other utterances in the corpus. This should be the word that has the lowest frequency in the rest of the corpus. We chose the most significant word approach to generate the default categories, because usually in human dialogues the intent of the speakers is hiding in the least-frequent, highest-information word. The program calculates the Afrikaans corpus word-frequency list, and then a comparison is run against each token in each pattern to find the least frequent word with that pattern. Four categories holding the most significant word were added to handle the positions of this word first, middle, last or alone. The feedback showed improvement in user satisfaction (Abu Shawar & Atwell 2003b).

A restructuring module was added in this version to map all patterns with the same response to one form, and to transfer all repeated patterns with different templates to one pattern with a list of alternative responses.

### 3.4 Learning from the Arabic and Arabic-English Qur'an

This version was updated to generate Arabic AIML files extracted from the Qur'an, the holy book of Islam. Moslems believe the Arabic text is a faithful transcription of the infallible words of God relayed through the angel Gabriel to the prophet Mohammed, who memorised the entire monologue to pass on verbally. Mohammed's successors transcribed the message to simplify trans-

---

```
<spreker2> is dit (lag) hoe gaan dit met Franna </spreker 2>
<spreker1> Franna </spreker1>
<spreker2> het Franna weer drie gedruk </spreker2>
<spreker1> nee hy't Donderdag twee gedruk <oorvleuel>
<spreker1> en Din~ </spreker1>
<spreker2> teen wie't hulle </spreker2> </oorvleuel> </spreker1>
<spreker2> gespeel </spreker2>
<spreker1> teen <fil> uh uuhm </fil> Proteapark </spreker1>
<spreker2> *a+ gewen </spreker2>
<spreker1> vyf-en-vyftig nul en Dinsdag het hulle agt-en-tagting nul gewen </spreker1>
```

---

Figure 3. Sample of spoken Afrikaans corpus

mission and avoid corruption, but every Moslem should aim to memorise it, in original Arabic, and to use the Qur'an to guide every aspect of their lives. The Qur'an consists of 114 sooras, which could be considered as sections, grouped into 30 parts (chapters). Each soora consists of more than one verse (Ayya). These ayyas are sorted, and must be shown in the same sequence. The AIML-learning system was revised to handle the non-conversational nature of the Qur'an. We assumed that if an input is an ayya, then the reply will be the next ayya. Children often learn the Qur'an in this way: the teacher cites an ayya, and the learner must recite the following ayya. So, our chatbot could be a novel tool to help learn the Qur'an. Two chatbot versions were created: the first accepts Arabic input and responds with the Arabic verse(s) (see Abu Shawar & Atwell 2004a). To help non-Arabic speakers (including one of the authors!) to understand the meaning of the interactions, the second version was retrained with a parallel Arabic-English version of the Qur'an; it also accepts English input and responds with both Arabic and English verse(s) (see Abu Shawar & Atwell 2004b). Figure 4 shows samples of the English and Arabic sooras of the Qur'an.

#### 4. Chatbot-Learning from the British National Corpus

It took several years for the Alice Botmaster to accumulate the 50,000 categories in the current public-domain set of AIML files (Wallace 2003). We wanted to investigate the possibility of using machine learning to extract a much larger set of AIML files: in theory, the chatbot-learning program can learn millions of categories given an appropriate dialogue corpus. We selected the BNC cor-

## THE DAYBREAK, DAWN, CHAPTER NO. 113

With the Name of Allah, the Merciful Benefactor, The Merciful Redeemer

113/1 Say: I seek refuge with the Lord of the Dawn

113/2 From the mischief of created things;

113/3 From the mischief of Darkness as it overspreads;

113/4 From the mischief of those who practise secret arts;

113/5 And from the mischief of the envious one as he practises envy.

(113) قلْ هُنَّا قُرُونٌ  
بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ  
الْقَلْقَلُ قَلْ أَعُوذُ بِرَبِّي  
مِنْ شَرِّ مَا خَلَقَ  
وَمِنْ شَرِّ<sup>1</sup> غَلِيقٍ إِذَا وَقَبَ  
وَمِنْ شَرِّ الْفَقَاثَاتِ فِي الْعُقَدِ  
وَمِنْ شَرِّ حَامِدٍ إِذَا حَسَدَ

Figure 4. Samples of the Arabic and English versions of the Qur'an

pus to train our program, the largest dialogue corpus readily available. Abu Shawar and Awell (2005) present two uses of the BNC: to automatically generate the largest AIML model ever; and to use chatbots trained on specific subsets of the BNC to "animate" or illustrate the type of English used with a specific domain or speaker-type. The British National Corpus (BNC 2002) is a collection of text samples amounting to over 100 million words, extracted from 4124 modern British English texts of all kinds, both spoken and written. The corpus is annotated using SGML (XML-like) mark-up, including CLAWS Part-of-Speech category of every word. All annotations are marked between <angle brackets>. The corpus is partitioned into two types: the spoken and the written transcripts. Herring (1996) argues that computer mediated communication (CMC) "is typed, and hence like writing, but exchanges are often rapid and informal, and hence more like spoken conversation", and Grondelaers et al. (2003) describe the language of Internet Relay Chat (IRC) as an example of "spoken language in written form"; so we decided to retrain ALICE using the BNC spoken transcripts.

#### 4.1 The BNC spoken dialogue transcripts

The spoken dialogue transcripts amount to 10 million words, and can be divided into two parts: a demographic part, involving transcriptions of spontaneous natural conversations between families, friends, and so forth, and the context-governed part, containing transcriptions recorded in educational, in-

formative, business, leisure, institutional, and public events (Crowdy 1994). Each corpus file starts with a long Header section, containing details of source, speakers, etc. In the transcript Body, the dialogue consists of a series of utterances or speaker-turns, marked at start and end by <u> and </u> tags. Each utterance tag also includes a speaker number (anonymised, eg F72PS002). Within a text sample, all sentences are tagged <s> and numbered; and each word is preceded with a CLAWS Part-of-Speech tag, e.g. ITJ = interjection, PUN = punctuation-mark, NP0 = singular proper name. An example of a sequence of two utterances is:

```
<u who=F72PS002>
< s n="32"><w ITJ>Hello<c PUN>.
</u>
<u who=PS000>
< s n="33"><w ITJ>Hello <w NP0>Donald<c PUN>.
</u>
```

Stripped of XML markup, this is simply an opening to a conversation:

F72PS002: Hello  
PS000: Hello Donald

The corresponding AIML atomic category can be generated:

```
<category>
<pattern>HELLO</pattern>
<template>Hello Donald</template>
</category>
```

However, the translation process from BNC format to AIML is not as simple as it might seem to be on the surface. A range of problems emerged during the translation process, which will be discussed in the following subsections.

#### 4.1.1 More than two speakers

Since the number of participants in chatbot dialogue is two, the user and the program, a dialogue corpus recorded between two parties would be most appropriate; we could then train the chatbot to mimic the part of one or other of the participants. However, the BNC recorded conversations covered a wide range of domains and often involved more than two speakers. When there are several dialogue participants, we cannot simply identify one participant as taking the place of the chatbot, so we cannot just follow one speaker in training. Instead, we assume that every utterance, by any speaker, is a candidate pat-

tern, and the subsequent utterance, regardless of speaker, is the corresponding template. The resultant AIML merges the contributions of all speakers.

#### **4.1.2 Unclear sections in utterances**

Since all spoken samples are transcribed from recorded speech, some parts of the utterances are unclear. The BNC uses the <unclear> tag to mark these, as in the following example:

```
<u who=PS000>
<s n="5"><unclear> <w AT0>a <w NN1>minute<c PUN>.
</u>
<u who=PS100 ><unclear ></u>
<u who=F72PS000>
<s n="6"><w CJC>And <w DTQ>what <w VBB>are <w PNP>they<c
PUN>?
</u>
```

Stripped of XML markup, this becomes:

```
PS000: ??? a minute
PS100: ???
F72PS000: And what are they?
```

A problem with the unclear turn is that it might be a response to a previous utterance, or it might introduce a new idea, which the next speaker responds to. In the translation to AIML we cannot decide if the unclear turn is a pattern or a template. To solve this problem we decided to remove the unclear turns. There are two approaches to elimination, either before or after the converter module maps pairs of successive utterances into pattern+template categories. The difference between them is as follows.

Assume that there are four speakers denoted by (spk) and the sequence of turns is: spk1→ spk2→ unclear→ spk3→ spk4.

The first approach is to omit the unclear turn itself before converting the transcript. In this case we will have the following sequence of utterances: spk1→ spk2→ spk3→ spk4. The conversion process will generate three categories: (spk1→ spk2), (spk2→ spk3), (spk3→ spk4).

The second approach is to omit the unclear after the conversion. After considering each pair as a pattern and a template, we have: (spk1, spk2), (spk2, unclear), (unclear, spk3), (spk3, spk4). Then any pair containing the unclear is excluded, so we will have two categories left.

The second approach completely sidesteps the problem by deleting the two cases where the unclear is a pattern or a template, and this means avoiding the category where spk3 is a response to spk2, which did not actually happen during the conversation. However, it is arguably possible to consider spk3 as a possible response to spk2, even if it did not really happen in this sequence, as at least the utterance is still a continuation of the conversation. As our goal was to generate a large set of categories automatically, we decided to adopt the first approach.

#### 4.1.3 Overlapping utterances

Overlapping represents the case where more than one speaker was active at the same time; this occurs during human conversation but not during chat-bot interactions. The BNC corpus transcribed the overlapping turns using an alignment map tag `<align>` to synchronise points within a spoken text, declared at the start of the division or text concerned; and the pointer tag `<ptr target=>` points to the identifier which was synchronised. The following example illustrates this problem:

```
<u who=w0014>
<s n=00011><w AJ0>Poor <w AJ0>old <w NP0>Luxembourg' <w VBZ>s
<w AJ0-VVN>beaten<c PUN>.
<s n=00012><w PNP>You <w PNP>you<w VHB>'ve <w PNP>you<w
VHB>'ve
<w AV0>absolutely <w AV0>just<w VVN>gone <w AV0>straight
<ptr target=P1> <w PRP>over <w PNP>it <ptr target=P2> </u>
<u who=w0001>
<s n=00013><ptr target=P1> <w PNP>I <w VHB>haven<w XX0>'t<c
PUN>.
<ptr target=P2> </u>
<u who=w0014>
<s n=00014><w CJC>and <w VVN>forgotten <w AT0>the <w AJ0>poor
<w AJ0>little<w NN1>country<c PUN>. </u>
```

The equivalent in a more human-readable format is:

W0014: Poor old Luxembourg's beaten. You, you've, you've absolutely just gone straight over it ...  
W0001: (interrupting) I haven't.  
W0014: ... and forgotten the poor little country.

The overlapping interruption problem is similar to the unclear section problem: both could impinge on the dialogue and affect what was said afterwards, but both require special handling to map onto pattern+template pairs. In our earlier system, learning from French and Afrikaans corpora, we simply ignored the all utterances involved, i.e. the interruption and also the interrupted utterances (the Qur'an does not have overlaps and interruptions, avoiding this problem for our Arabic chatbot). This meant we lost potential categories; the above example would just be skipped. For the BNC learning model, since the overlapping turn is separated, we treat it as a new turn; this gives us two pattern+template categories for the above example.

#### 4.1.4 Using abbreviations

The encoders used some enclitics in writing the recorded speech as: “I’d”, “he’ll”, “John’s”, and so on. A problem arises in converting such abbreviations to the AIML patterns. We have to remove all punctuations from the pattern to be accepted by the ALICE interpreter. To date our machine-learnt models have not included linguistic analysis markup, such as grammatical, semantic or dialogue-act annotations (Atwell 1996; Atwell et al. 2000a, b), as ALICE/AIML makes no use of such linguistic annotations in generating conversation responses. It cannot distinguish if “s” is an abbreviation of “is” or “has” or a possessive. We decided to remove all punctuations without expanding the enclitic. Even though a sentence such as “I’d like” will be mapped into “I d like”, this is still compatible with our approach of the most significant word, since whether “d” denotes had or would, it will not be the most significant word in the sentence. The “n’t” abbreviation was the only one replaced by “not”, so “haven’t” becomes “have not” instead of being “haven t” which is a different word with different meaning, which might be erroneously selected as the most significant word.

#### 4.1.5 Using character entity references

Some transcripts included foreign words including accented characters, encoded using HTML character entity references, such as “&agrave;”, “&Ouml;”, and so on. Unfortunately these non-standard letters raised problems during compilation of the AIML files, and furthermore could not match input from a UK English keyboard. So, all entity references were replaced with the corresponding unaccented characters. For example: “&agrave;” is mapped to “a”, and “&Ouml;” to “O”.

#### 4.1.6 Linguistic annotations

The spoken transcripts include markup of paralinguistic phenomena such as: voice quality (whispering, laughing, etc.), non-verbal but vocalised sounds (coughs, humming noises), non-verbal and non-vocal events (animal noises, passing lorries), significant pauses (silence) and speech management phenomena (truncation, false starts). These phenomena might be of interest for other purposes, but the auditory features will not occur when chatting with a computer via keyboard and screen text. So we removed all linguistic annotations including the POS tags. For example:

```
<u who=PS21K>
<s n="37"><w CRD>forty <w NN0>percent <w PRF>of <w DPS>her
<w NN1>time
<w CJS>because <w PNP>she <w VDZ>does <w PNP>it <w AV0>so
<w AV0>quickly <vocal desc=laugh> <w CJC>but <w UNC>er <w
ITJ>oh </u>
```

The utterance stripped of XML markup will be:

PS21K: forty percent of her time because she does it so quickly but er oh

#### 4.1.7 Long monologues

We followed the BNC partitioning into utterances, even though sometimes the transcribers marked an utterance as running over several sentence-boundaries. For example:

```
<u who=F72PS000>
<s n="29"><w PNP>You <w VDB>do<c PUN>?
<s n="30"><w AV0>Well <w PNP>you <w VBB>are <w AV0>very <w
AJ0>fortunate <w NN0>people<c PUN>.
<s n="31"><w CJC>But <w PNI>none <w PRF>of <w PNP>you <w
VM0>will <w VVI>know <w DPS>my <w NN1>friend <w AV0>over
here <w DTQ>whose <w NN1>name <w VBZ>is <w NP0>Donald<c
PUN>. </u>
```

Stripped of XML markup, this is equivalent to:

F72PS000: You do? Well you are very fortunate people. But none of you will know my friend over here whose name is Donald.

The program merges all of the sentences to form one string starting with <u who..> and ends with </u>. This generates a very long turn; this is not normally found in computer-human chatting. The alternative would be to artificially treat each <s> as a separate turn, splitting the above into 3 pseudo-utterances; but we decided that, as our aim is to investigate the utility of a corpus for machine-learning, we should follow the boundaries set out in the corpus rather than reinterpret them.

## 4.2 Adapting the learning software to the BNC

We modified the Afrikaans.java system to cope with the BNC samples:

1. Using the lemmatised BNC frequency list (Kilgarriff 1996) in extracting the least frequent words.
2. Modifying the algorithms to handle the BNC-specific annotations and problems discussed above.
3. The large AIML file learnt from the BNC proved too big for the default ALICE engine to handle, so we had to find a work-around.

### 4.2.1 *The BNC frequency list*

“A central fact about a word is how common it is. The more common it is, the more important it is to know it.” (Kilgarriff 1996: 135). Kilgarriff argues that language learners should be taught the commonest words first, so they understand them and know how to use them. Kilgarriff echoed Zipf’s observation that the most common words dominate real use.

Kilgarriff extracted two word-frequency lists from the BNC, the lemmatised and unlemmatised list. The lemmatised frequency list includes 6,318 words with more than 800 occurrences in the whole 100-million-word BNC. The frequency of verbal words and its nominal are generated separately, where the count of the verb is the sum of counts of all instances for each verbal, so the frequency of verbal ‘aim’ will count ‘aims’, ‘aiming’, and ‘aimed’. In contrast, the unlemmatised list counts the frequency for each verb-form separately. The unlemmatised list gives the frequency, the word, the PoS, and finally the number of files the word occurs in, as illustrated below:

6187267	<i>the</i>	at0	4120
2941444	<i>of</i>	prf	4108
2682863	<i>and</i>	cjc	4120
2126369	<i>a</i>	at0	4113
1812609	<i>in</i>	prp	4109

The program starts by reading the frequency list and mapping it into a vector named “bnc\_freq”. The next step is to read the file name from the index and adding all utterances into a vector named “dialogue”. Now the same phases of version (2) are used as follows: the dialogue vector elements are filtered, reiterated, and prepared to originate pattern and template sequentially. Then the dialogue vector is re-structured where all different patterns with the same template are categorised as <srai> categories, and all different templates related to the same pattern are grouped as an atomic category with random list. After that all categories are copied into an AIML file. Finally the process is repeated again by accessing the index and selecting the next transcript to be read.

The reading process involves two aspects:

1. Extracting the word and its frequency, disregarding the POS and the number of files in which the word occurs.
2. Ignoring numbers and any non-orthographic words such as “in-spite-of”; non-orthographic words will not be found in the AIML pattern, especially after removing all punctuations.

The extracted pair <word, frequency> is inserted into the “bnc\_freq” vector. The vector will be used later on to obtain the frequency of each token in the pattern.

Some BNC spoken tokens were not found in the unlemmatised list, such as “huhuhuhu”; in such cases the token itself is considered as the least frequent word. Since the BNC spoken transcripts are annotated with part-of-speech tags, we used these tags to filter the meaningful words to be used as the first word or least frequent words: *wh-question-words*, prepositions, and pronouns are not considered. This modification improves the matching process and we record better user satisfaction than before.

#### 4.2.2 Text normalization for BNC files

The BNC-specific format used to annotate dialogues required changes in the filtering process, including removal of unnecessary linguistic annotations. We modified the normalisation module as follows:

1. Removing the unclear turns.
2. Deeming the overlapping turns as separate ones. The overlap is referenced as an individual turn in the BNC corpus, and since we want to maximise the number of categories, we consider it as a turn rather than eliminating it as in earlier versions of the program.
3. Replacing enclitics and abbreviations, e.g. “n’t” with “ not”.
4. Replacing the character entity references with normal alphabetic characters.

The preparation phase began by considering the first element in the vector as a pattern and the second as a template. After removing all punctuation from the pattern, the first word of each pattern is used to create a new default category holding the first word followed by star, which represents the first word approach. After that, the pattern is tokenised, and the “bnc\_freq” vector generated in module one is scanned to extract the frequency for each token in the pattern. The generated list is sorted by frequency in ascending order, and the first token is considered the most significant word (least frequent one). The process continues by generating four categories: atomic category holding the least frequent word only, and another three default categories holding the least frequent word in the first, middle, and last of the sentence. Then the restructuring phase is executed and the final categories are written to an AIML file.

#### *4.2.3 The problems in scaling up ALICE to very large AIML files*

During the program run, the execution terminated many times due to an “out of memory storage” problem. This problem related to the large size of some files, around 2MB. To solve this problem, the large files were distributed into several smaller files. This allowed us to load up the BNC-learnt AIML into an ALICE chatbot hosted at the Padorobot website; but this proved to be only a temporary solution, see below.

### **4.3 Results and evaluation**

After nearly ten days of running the program, 1,153,129 categories were generated. This number is 20 times bigger than any existing chatbot: the large public-domain AIML category-file developed over several years (Wallace 2003) contains about 50,000 categories.

We tried to create a new version of ALICE using the Pandorabot web-hosting service (Pandorabot 2003) to make our chatbot available for use over the World Wide Web. After the successful compilation, and a little chatting, we received a message from the Pandorabot that the big size of our dialogue model was causing major problems for the server and to other user chatbots; so we had to take the BNC version offline again.

The evaluation of the AIML-learning system can be tackled into three directions:

1. Dialogue efficiency in terms of matching type.
2. Dialogue quality metrics based on response type.
3. Users' satisfaction assessment based on an open-ended request for feedback.

From the dialogue efficiency and quality we aim to measure the success of the first and most significant word approaches. We had previously measured the quality and efficiency of the Afrikaans dialogue generated between users and Afrikaans chatbot version (for more details see Abu Shawar & Atwell 2003b); we concluded that the most significant word approach increases the ability of finding answers to general user input. In order to measure the quality of each response, we classified the responses into three types: reasonable, weird but reasonable, or nonsensical. A human Afrikaans speaker and teacher evaluated the dialogue; for this evaluator, it seemed that "nonsensical" responses are more likely than reasonable or understandable but weird answers.

User satisfaction plays a critical point in evaluation; following feedback from users who tried chatting with the Afrikaans version, we expanded the AIML pattern matching using the first and least-frequent word approaches.

Using the BNC version, we relied on user satisfaction to evaluate the system. The first impressions from users were disappointing; Chatting (3) is a sample of dialogue between a user and the chatbot where we tried to analyse the reasons behind users' disappointment.

---

**Chatting (3). A sample of chatting with the BNC chatbot version**

---

User: Hi

Chatbot: good morning. Now young lady what can I do for you today?

User: Let's talk.

Chatbot: Yeah

User: Do you like dancing?

- Chatbot: No. I ve. I ve just. All I ve done is because Erm because that is a function of that.
- User: Tell me about dancing:
- Chatbot: The dances. Oh they were just great fun the dances all met in the school. It was the nearest to us and the next one was the Glen Ayloch school. But oh we d we d a lot of fiddlers in those days and a girl at the Piano. And then when things got a wee bitty better we used to have the Camerons up from Kilray from Kerry to dance to the. Play at the dances. And of course the night of the. Gathering Ball it was just packed tight with people hardly, the school would hardly hold them all. And then they put up a big tent in the. Beside the school where you went out for supper er it was that S s.
- User: You had a dancing party at your school?
- Chatbot: That s good.
- User: I like swimming
- Chatbot: Mhm.
- User: Do you like swimming?
- Chatbot: No. I ve. I ve just. All I ve done is because Erm because that is a function of that.
- User: Talk to you later.
- Chatbot: Oh. It is still fine.
- User: Bye bye
- Chatbot: Right cheerio now.

A careful look at this example dialogue provides us with an insight into some remaining problems with the pattern matching process. Some responses seem meaningless, because the response is taken from a specific BNC text and this chat is not in the same context. Other responses show that we need to improve our pattern-extraction process. For example, when the user input starts with “do you like...”, the same response will be generated regardless of the object. This is because it matches the first word approach before the most significant one. To solve these problems we improved the first word approach by considering the first word scanned in the pattern, which is not a question, a pronoun, or a preposition. The first “meaningful” word in the utterance will be considered instead; also the least “meaningful” word will be considered as least frequent word. To concentrate on certain contexts, we extracted specific domains from

**Table 2.** BNC subdomain specialist versions

Chatbot	Atomic		Default		Total Categories
	Categories	File size (KB)	Categories	File size (KB)	
Sport	3283	974	7913	1,820	11196
World affairs	3120	983	8756	1,886	11876
Travel	640	314	1636	575	2276
Media	1061	491	3126	1,210	4187
Food	503	93	1125	168	1628

**Table 3.** The BNC-trained London Teenager and Loudmouth chatbots

Chatbot version	Atomic		Default		Total Categories
	Categories	File size (KB)	Categories	File size (KB)	
Michael	7021	1,044	14914	1,905	21935
Jessica	4261	787	10069	1,554	14330
Peter	5912	939	12878	1,771	18790
Robin	4726	673	14580	1,774	19306
Samantha	6058	788	18283	1,695	24341
Loudmouth1	8114	1,301	17827	2,368	25941
Loudmouth2	8383	1,217	16971	2,130	25354

the BNC and used them to develop different versions of the chatbot as shown in Table 2.

Table 3 lists another five chatbots generated from London teenager transcripts, trained to talk like London teenagers; and two “Loudmouth” chatbots, trained with “interesting” BNC files containing exemplary swearing, as suggested by participants in the CORPORA e-mail discussion list.

Chatting (4) illustrates the dialogue between a user and the Peter chatbot.

#### Chatting (4). A sample of dialogue with Peter chatbot version

- Human: hello
- Peter: Hello, hello see it sounds sort of like like you’re listening to the tape, but it
- Human: do you like music?
- Peter: Oh are you going home or are you gonna be back?
- Human: I’m going home, what about you?
- Peter: Oh well they’re bloody playing football again are not they, fucking pooters No, I bet they’re playing football with the dikes

---

Human: are you going to watch a football game?  
Peter: Are you sure?  
Human: I'm asking?  
Peter: Erm  
Human: what about football?  
Peter: Fuck off  
Human: This is rude.  
Peter: Yes of course yes Right

---

## 5. Conclusions

Software capable of imitation of human communication behaviour is a fundamental goal of Natural Language Processing (NLP) research. This goal has led researchers to develop machine conversations systems, or chatbots. However, most chatbots require linguistic knowledge to be hand coded, and are restricted to the language spoken by the developer. We developed a java program that uses a corpus-based approach to train ALICE. The program was used to create different versions of ALICE speaking varieties of English, French, Afrikaans, and Arabic languages. We modified the program to use the British National Corpus spoken transcripts. We were able to develop two learning techniques, the first word and the most significant word approaches, which were successful in learning 1,153,129 categories extracted from the BNC corpus. Two goals were achieved from the automation process: the possibility of generating different versions in different languages, bringing chatbot technology to languages with few if any NLP resources; and the ability to learn a very large number of rules (categories) within a short time, saving effort and errors in doing such work manually. The conversation rules are automatically derived from text, without need of mark-up or linguistic tagging; for example the PoS tagging or socio-linguistic speaker information in BNC files was not needed or used. This means chatbots can be derived from any dialogue transcripts, even untagged corpora. Our Afrikaans chatbot has been acknowledged at Potscheftsroom University as a groundbreaking example of emerging Afrikaans NLP technology; and our BNC-trained chatbot has learnt a set of rules which is larger than any other NLP knowledge base, and is probably the largest AI rule-based system ever.

It is hard to evaluate “accuracy” or “relevance” of chatbot responses, since there is no simple automated metric of “relevance”. All our chatbots were made

available on the Padorabots.com website for public access and testing, and we elicited some subjective feedback from users. The Afrikaans and Arabic Qur'an chatbots drew mainly favourable feedback; but the London Teenager and Loudmouth chatbots seemed to impress less, users found some responses not just rude but incoherent. Perhaps one lesson is that corpus-trained chatbots should be seen to be "useful" to be appreciated.

## References

- Abu Shawar, B. & Atwell, E. (2005). A chatbot system as a tool to animate a corpus. *ICAME Journal*, 29, 5–24.
- Abu Shawar, B. & Atwell, E. (2004a). An Arabic chatbot giving answers from the Qur'an / *Un chatbot arabe qui donne des réponses du Coran*. In B. Bel & I. Marlien (Eds.), *Proceedings of TALN2004: XI Conference sur le Traitement Automatique des Langues Naturelles* (Volume 2, pp. 197–202). ATALA.
- Abu Shawar, B. & Atwell, E. (2004b). Accessing an Information system by chatting. In F. Meziane & E. Metais (Eds.), *Natural Language Processing and Information Systems: Proceedings of NLDB04* (pp. 407–412). Berlin: Springer-Verlag.
- Abu Shawar, B. & Atwell, E. (2003a). Using dialogue corpora to retrain a chatbot system. In D. Archer, P. Rayson, A. Wilson & T. McEnery (Eds.), *Proceedings of the Corpus Linguistics 2003 conference (CL2003)* (pp. 681–690), UCREL Technical Paper 16. Lancaster: Lancaster University.
- Abu Shawar, B. & Atwell, E. (2003b). Using the Corpus of Spoken Afrikaans to generate an Afrikaans chatbot. *SALALS Journal: Southern African Linguistics and Applied Language Studies*, 21, 283–294.
- Abu Shawar, B. & Atwell, E. (2003c). Machine Learning from dialogue corpora to generate chatbots. *Expert Update Journal*, 6 (3), 25–30.
- Abu Shawar, B. & Atwell, E. (2002). *A comparison between Alice and Elizabeth chatbot systems*. School of Computing research report 2002.19. Leeds: University of Leeds.
- ALICE (2002). *A.L.I.C.E AI Foundation website* (<http://www.Alicebot.org/> or <http://Alicebot.franz.com/>)
- Atwell, E. (1996). Machine Learning from corpus resources for speech and handwriting recognition. In J. Thomas & M. Short (Eds.), *Using corpora for language research: Studies in the honour of Geoffrey Leech* (pp. 151–166). Harlow: Longman.
- Atwell, E. (1988). Transforming a Parsed Corpus into a Corpus Parser. In M. Kytö, O. Ihälainen & M. Risanen (Eds.), *Corpus Linguistics, Hard and Soft: Proceedings of the ICAME 8th International Conference on English Language Research on Computerised Corpora* (pp. 61–70). Amsterdam: Rodopi.
- Atwell, E. (1987). How to detect grammatical errors in a text without parsing it. In B. Maegaard (Ed.), *Proceedings of EACL: the Third Conference of European Chapter of the Association for Computational Linguistics* (pp. 38–45). New Jersey: ACL.
- Atwell, E. (1983). Constituent Likelihood Grammar. *ICAME Journal*, 7, 34–67.

- Atwell, E. & Drakos, N. (1987). Pattern Recognition Applied to the Acquisition of a Grammatical Classification System from Unrestricted English Text. In B. Maegaard (Ed.), *Proceedings of EACL: The Third Conference of European Chapter of the Association for Computational Linguistics* (pp. 46–54). New Jersey: ACL.
- Atwell, E., Howarth, P. & Souter, C. (2003). The ISLE corpus: Italian and German spoken learner's English. *ICAME Journal*, 27, 5–18.
- Atwell, E., Demetriou, G., Hughes, J., Schiffрин, A., Souter, C. & Wilcock, S. (2000a). A comparative evaluation of modern English corpus grammatical annotation schemes. *ICAME Journal*, 24, 7–23.
- Atwell, E., Howarth, P., Souter, C., Baldo, P., Bisiani, R., Pezzotta, D., Bonaventura, P., Menzel, W., Herron, D., Morton, R. & Schmidt, J. (2000b). User-Guided System Development in Interactive Spoken Language Education. *Natural Language Engineering journal: Special Issue on Best Practice in Spoken Language Dialogue Systems Engineering*, 6 (3–4), 229–241.
- Atwell, E., Leech, G. & Garside, R. (1984). Analysis of the LOB Corpus: progress and prospects. In J. Aarts & W. Meijis (Eds.), *Corpus Linguistics: Proceedings of the ICAME 4th International Conference on the Use of Computer Corpora in English Language Research* (pp. 40–52). Amsterdam: Rodopi.
- Batacharia, B., Levy, D., Catizone, R., Krotov, A. & Wilks, Y. (1999). CONVERSE: a conversational companion. In Y. Wilks (Ed.), *Machine conversations* (pp. 205–215). Boston/Dordrecht/London: Kluwer.
- BNC (2002). *British National Corpus website* (<http://www.natcorp.ox.ac.uk/>)
- Colby, K. (1999). Human-computer conversation in a cognitive therapy program. In Y. Wilks (Ed.), *Machine conversations* (pp. 9–19). Boston/Dordrecht/London: Kluwer.
- Colby, K. (1973). Simulation of belief systems. In R. Schank & K. Colby (Eds.), *Computer models of thought and language* (pp. 251–286). San Francisco: Freeman.
- Crowdy, S. (1994). Spoken corpus transcription. *Literary and Linguistic Computing*, 9 (1), 25–28.
- Grondelaers, S., Speelman, D. & Geeraerts, D. (2003). A corpus-based approach to informality: The case of Internet chat. In D. Archer, P. Rayson, A. Wilson & T. McEnery (Eds.), *Proceedings of the Corpus Linguistics 2003 conference (CL2003)* (p. 264). UCREL Technical Paper 16. Lancaster: Lancaster University.
- Herring, S. (1996). Introduction. In S. Herring (Ed.), *Computer-mediated communication: linguistics, social and cross-cultural perspectives* (pp. 1–10). Amsterdam: John Benjamins.
- HEXBOT (2004). *HEXBOT chatbot website*. (<http://www.hexbot.com/>)
- Hughes, J. & Atwell, E. (1994). The automated evaluation of inferred word classifications. In A. Cohn (Ed.), *Proceedings of ECAI'94: 11th European Conference on Artificial Intelligence* (pp. 535–540). Chichester: John Wiley.
- Hutchens, J. (1996). *How to pass the Turing test by cheating*. School of Electrical, Electronic and Computer Engineering research report TR97-05. Perth: University of Western Australia.
- Jurafsky, D. & Martin, J. (2000). *Speech and Language Processing*. Prentice Hall.
- Kerr, B. (1983). *Minnesota Corpus*. Minneapolis: University of Minnesota Graduate School.

- Kilgarriff, A. (1996). Putting Frequencies in the Dictionary. *International Journal of Lexicography*, 10 (2), 135–155.
- Mann, W. (2002). *Dialog Diversity Corpus website*. (<http://www-rcf.usc.edu/~billmann/diversity/DDivers-site.htm>)
- Nelson, G. (2002). *International Corpus of English: The Singapore Corpus user manual*. ([http://www-rcf.usc.edu/~billmann/diversity/ICE-SIN\\_Manual.PDF](http://www-rcf.usc.edu/~billmann/diversity/ICE-SIN_Manual.PDF))
- Pandorabot (2003). *Pandorabot chatbot hosting website*. (<http://www.pandorabots.com/pandora>)
- Saygin, A., Cicekli, I. & Akman, V. (2000). Turing test: 50 years later. *Minds and Machines*, 10 (4), 463–518.
- Van Rooy, B. (2003). *Transkripsiehandleiding van die Korpus Gesproke Afrikaans (Transcription Manual of the Corpus of Spoken Afrikaans)*. Potchefstroom: Potchefstroom University.
- Wallace, R. (2003). *The elements of AIML style*. ALICE AI Foundation.
- Weizenbaum J. (1967). Contextual understanding by computers. *Communications of the ACM*, 10 (8), 474–480.
- Weizenbaum, J. (1966). ELIZA-A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 10 (8), 36–45.

#### *Author's address*

Bayan Abu Shawar and Eric Atwell  
School of Computing, University of Leeds  
Leeds LS2 9JT, England  
[bshawar@comp.leeds.ac.uk](mailto:bshawar@comp.leeds.ac.uk)  
[eric@comp.leeds.ac.uk](mailto:eric@comp.leeds.ac.uk)

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/343812815>

# Memory-Based Deep Neural Attention (mDNA) for Cognitive Multi-Turn Response Retrieval in Task-Oriented Chatbots

Article in Applied Sciences · August 2020

DOI: 10.3390/app10175819

---

CITATIONS

5

READS

261

3 authors, including:



Obinna Agbodike

NAU

6 PUBLICATIONS 10 CITATIONS

[SEE PROFILE](#)



Lei Wang

AKKA Technologies

175 PUBLICATIONS 2,860 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



anomalous hall effect [View project](#)

Article

# Memory-Based Deep Neural Attention (mDNA) for Cognitive Multi-Turn Response Retrieval in Task-Oriented Chatbots

Jenhui Chen <sup>1,2,3,4,†</sup>, Obinna Agbodike <sup>5</sup> and Lei Wang <sup>6,\*</sup>

<sup>1</sup> Department of Computer Science and Information Engineering, Chang Gung University, Kweishan, Taoyuan 33302, Taiwan; jhchen@mail.cgu.edu.tw

<sup>2</sup> Artificial Intelligence Research Center, Chang Gung University, Kweishan, Taoyuan 33302, Taiwan

<sup>3</sup> Center for Artificial Intelligence in Medicine, Chang Gung Memorial Hospital, Kweishan, Taoyuan 33375, Taiwan

<sup>4</sup> Department of Electronic Engineering, Ming Chi University of Technology, Taishan District, New Taipei City 24301, Taiwan

<sup>5</sup> Department of Electrical Engineering, Chang Gung University, Kweishan, Taoyuan 33302, Taiwan; d0721009@cgu.edu.tw

<sup>6</sup> School of Software, Dalian University of Technology, Dalian 116024, China

\* Correspondence: lei.wang@dlut.edu.cn

† This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2221-E-182-042-MY2; and in part by the Chang Gung Memorial Hospital, Kweishan, Taoyuan, Taiwan, under Grants CMRPD2J0012 and CMRPD2I0052.

Received: 28 July 2020; Accepted: 21 August 2020; Published: 22 August 2020



**Abstract:** One of the important criteria used in judging the performance of a chatbot is the ability to provide meaningful and informative responses that correspond with the context of a user's utterance. Nowadays, the number of enterprises adopting and relying on task-oriented chatbots for profit is increasing. Dialog errors and inappropriate response to user queries by chatbots can result in huge cost implications. To achieve high performance, recent AI chatbot models are increasingly adopting the Transformer positional encoding and the attention-based architecture. While the transformer performs optimally in sequential generative chatbot models, recent studies have pointed out the occurrence of logical inconsistency and fuzzy error problems when the Transformer technique is adopted in retrieval-based chatbot models. Our investigation discovers that the encountered errors are caused by information losses. Therefore, in this paper, we address this problem by augmenting the Transformer-based retrieval chatbot architecture with a memory-based deep neural attention (mDNA) model by using an approach similar to late data fusion. The mDNA is a simple encoder-decoder neural architecture that comprises of bidirectional long short-term memory (Bi-LSTM), attention mechanism, and a memory for information retention in the encoder. In our experiments, we trained the model extensively on a large Ubuntu dialog corpus, and the results from recall evaluation scores show that the mDNA augmentation approach slightly outperforms selected state-of-the-art retrieval chatbot models. The results from the mDNA augmentation approach are quite impressive.

**Keywords:** Bi-LSTM; memory; NLP; attention; dialog-system; retrieval

## 1. Introduction

Many studies on natural language processing (NLP) agree that chatbots have contributed immensely towards the advancement in information retrieval and exchange between humans and computers. In the vertical domain, the profitability of online transactions in e-commerce, reservations, and marketing firms, etc., is beginning to depend heavily on chatbots to convince people to make

purchase of goods and services through interactive and persuasive chats. For this reason, it is important that a chatbot is able to respond coherently and accurately with respect to the context of queries from users. Therefore, we opine that the degree of the accuracy of selected response is the primary criteria to be used in judging the performance of a chatbot. However, modeling a chatbot to consistently select and match accurate responses with respect to the intent and context of the users' input utterances over multiple-turns of a conversation is a challenging task.

To address this challenge, a plethora of research studies have been done on retrieval-based dialog systems. These systems are more commonly adopted in vertical domains unlike in open domains where sequence-to-sequence generative chatbots are more prevalently adopted. Retrieval chatbots enjoy the advantage of possessing rich repositories from which they can select information-rich responses [1,2] that align within the scope of a predefined knowledge base. Whereas the generative-based ones possess so much liberty on how to respond to utterances, and thus, are usually prone to occasionally generate grammatical errors or irrelevant generic responses that may not serve the intents of the user.

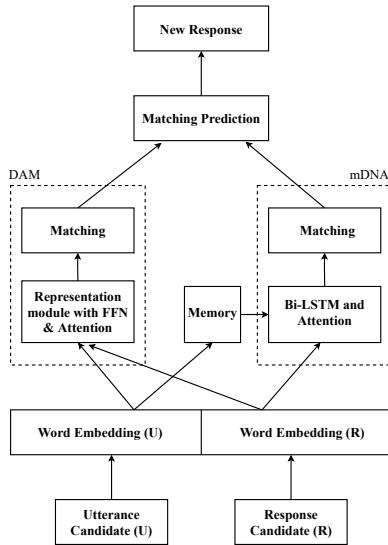
Early studies on retrieval chatbots considered only data in single-turn of a dialog for selecting responses from a repository [3–5]. In-other-words, this approach entails sole use of the information in the last conversational utterance for matching a response. Meanwhile, outcomes from recent works such as the deep attention matching network (DAM) by Zhou et al. [1], shows that the consideration of information in multiple-turns of dialog prove to offer better performance in-which response selection matching not only considers information in last utterance turn, but also considers previous turns of utterances for matching contextual dependencies, which significantly improve accuracy of output response selection. This approach is based on the notion that human conversational manner often depend on the context and sentiments in the previous turns of utterances containing varying segments of semantic cues.

Recently, most of the state-of-the-art multi-turn retrieval chatbot models adopt the novel Transformer architecture proposed by Vaswani et al. [6] because an attention-based network helps neural systems to expedite the capturing of utterance-response textual dependencies by replacing deep tensors, thus minimising computational cost [7]. However, Transformer-based retrieval chatbot models [1,2,8] encounter the problematic occurrence of the following errors:

- **logical inconsistency error:** where retrieved response candidates are wrong due to logical mismatch
- **vague or fuzzy response candidate error:** where selected responses contains improper details

To contribute towards solving the above outlined problems, we propose the memory-based deep neural attention (mDNA). The key purpose of the mDNA is to augment transformer-based retrieval chatbot models to improve response accuracy and logical consistency. The mDNA is a deep neural network (DNN) architecture with a Bi-LSTM encoder integrated with attention mechanism; and in addition, a memory module is implemented in the encoder. The purpose of implementing memory is to capture and store important utterance word embedding so as to offer a cognitive-like ability to the model. In our experiments, we adopted the DAM [1] as a case study representing the transformer-based retrieval chatbot model. We combined the two model architectures (i.e., DAM and mDNA) with an approach similar to unimodal late fusion method [9], as shown in Figure 1, to achieve enhanced performance.

During the response retrieval process, the previous multi-turns of relevant utterance information that could have probably been “forgotten” (i.e., data lost in DAM but retained in mDNA memory) are parsed through the mDNA Bi-LSTM encoder with the attention mechanism to match long-range contextual dependencies. The final response candidates from the DAM and mDNA are combined to undergo a voting process via a softmax function for the final response output selection. Our experiments show that the fusion of two different model architectures increases the number of response candidates. It also increases the probability of the chatbot to select more logically consistent and accurate responses. Thus, this technique drastically minimises the problematic occurrence of fuzzy response candidates and logical response inconsistency errors impairing the performance of most transformer-based retrieval dialog systems.



**Figure 1.** Operational flow of DAM augmented with mDNA.

## 2. Related Works

The growing number of research interests in multi-turn retrieval chatbots is an indication that it is a successful approach, but plagued with inherent challenges. Investigative studies done in [1,2,8] and so on, has sufficiently proven the superiority of multi-turn models over the single-turn ones. Therefore, with greater advantages, multi-turn retrieval chatbots are widely being adopted for various end-to-end task completion services [10] in the vertical domain which justifies their importance. Currently the existing efforts improve intuitive matching of utterance-context and semantic dependency to achieve high accuracy in response selection. It has driven recent scholarly works on retrieval chatbots [1,8,11,12] to adopt the use of Transformer attention-based architecture proposed in [6]. Although feed-forward neural networks (FFN) such as CNN are best suited for computer vision tasks, the Transformer attention model architecture has successfully used it for improving utterance textual context and response dependency matching in sequential neural dialog system. In addition, it has also achieved fair results in retrieval-based dialog system models such as the DAM network [1].

While the Transformer is now being hailed as the potential replacement to gated-recurrent neural networks in NLP tasks. Recent novel retrieval chatbot models based on the transformer-attention architecture [1,8,11] have shown to encounter the problematic occurrence of context mismatch and logical inconsistency errors in the selected responses of the models. Although some comparative studies on Transformer and RNN [13] agree that the self-attention and positional encoding attributes of the Transformer give them better performance leverage over the gate-based RNN neural models, more recent studies (e.g., [14]) point out that the multi-head attention mechanism of transformer models can also cause it to suffer loss of sequential information that are important in natural languages. Wang et al. [14] thereby proposed a RNN-enhanced Transformer (R-Transformer) model to address this problem.

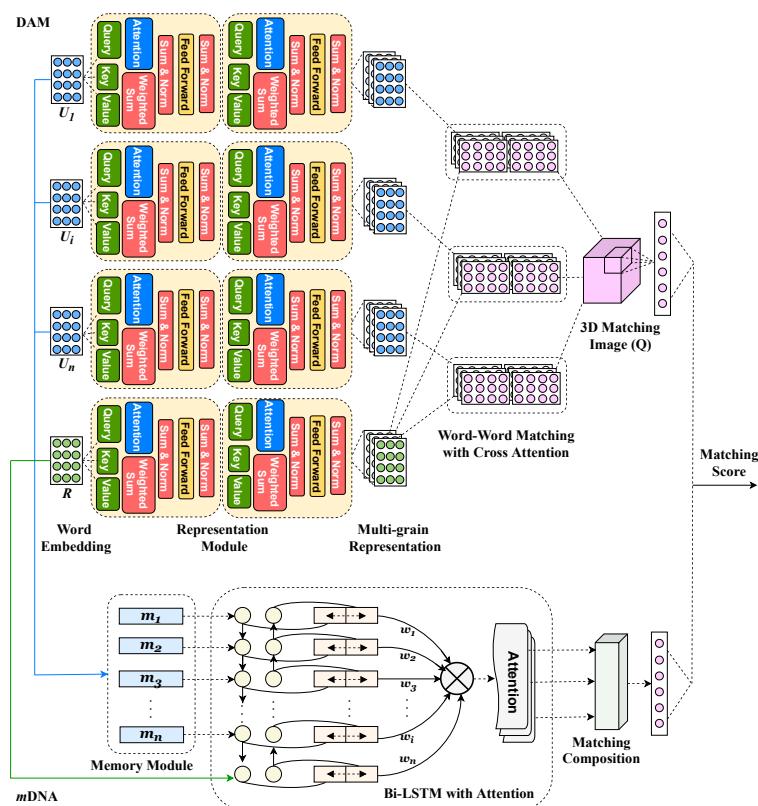
However, we argue that by systematically applying memory to gated RNN-based models, it would achieve comparable long-range contextual matching performance such as the Transformer without suffering significant loss of important data. Scholarly works that have investigated the integration of memory in gate-based recurrent neural models include Zhao et al. [11] who used background document to supply external knowledge to retrieval chatbot models (i.e., a memory-like operation) to enrich response coherence with respect to the context of utterances. In other studies, memory support has been added to neural networks for natural language transduction tasks in [15]. Also, in [16], an attempt has been made to replace the use of attention with active memory. Furthermore, Wulamu et al. [17] combined the use of memory networks and attention to achieve

improved responses on simple Q&A tasks but, in more complex Q&A tasks, their model performed poorly. Owing to the poor model performance of [17] in handling complex retrieval tasks, the authors consider implementing Bi-LSTM and attention mechanism in their future work. Finally, investigating the effectiveness of augmenting Transformer with RNN-based neural model to forge a single robust ensemble, the research by Amazon’s Domhan [18] suggests that one can successfully achieve good performance by such combination of architectures.

In this paper, we describe how we tackled the problems of logical mismatch and context inconsistency errors of multi-turn response selection in Transformer-based retrieval chatbots, by exploiting the collective benefits of memory, gated-RNN, and the Transformer attentional components combined as one unit. We selected the novel DAM network [1] to serve as a case study representing many other similar multi-turn retrieval chatbot models based on the Transformer architecture. By augmenting the DAM model with the mDNA, we achieved optimized performance, with respect to logical consistency and contextual accuracy of output responses.

### 3. Model Description

This section briefly describes the interactive roles of the major functional units that make up the mDNA (i.e., the Memory Encoding, Bi-LSTM with Attention, and the matching composition or prediction) as illustrated in Figure 2.



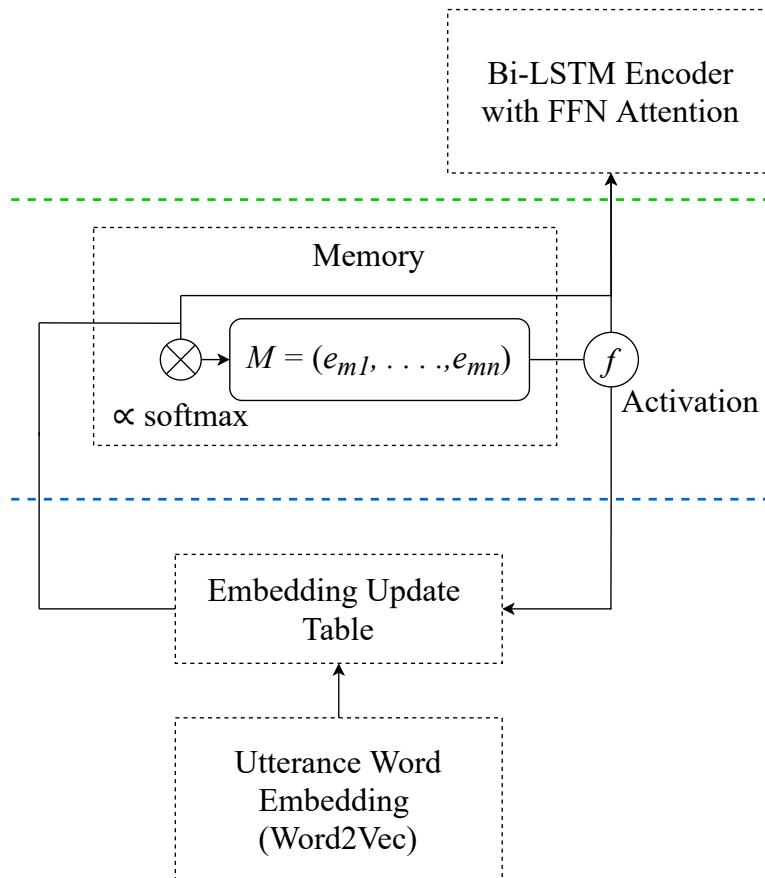
**Figure 2.** An overview of the combined architectures of DAM and mDNA.

#### 3.1. Memory Utterance Embedding

The Transformer-based models can effectively capture and match textual dependencies in utterance and response. However, important relevant word representations in long multi-turn dialogs can easily be lost in the process and thereby impair the quality of the final predicted output responses. On this basis, we formulated the concept of the mDNA to comprise a memory unit for temporary

storage of the utterance input so as to retain important contextual utterance information which will be needed consequently for response dependency matching during a conversational session.

Here, we denote the input utterance embedding parsed from word2vec to the memory as  $m = (m_1, m_2, m_3, \dots, m_n)$ . The response embedding is directly fed into the Bi-LSTM encoder and is denoted as  $r = (r_1, \dots, r_m)$ . Then using the pre-trained word embedding table, we generate two input sequences as  $M = [e_{m,1}, \dots, e_{m,n}]$  and  $R = [e_{r,1}, \dots, e_{r,m}]$ , where  $e \in R^d$ , and  $d$  is the dimension of word embedding. As shown in Figure 3, we use soft attention denoted as  $(\alpha_M)$  to filter important utterance words vectors of given lengths  $e_{ml}$  for storage. We express this as,  $\alpha_M = \text{softmax}(W, e_{m,l})$ , where  $W$  is the weight score, and  $e_{m,l}$  is utterance vector representation ( $m$ ), and the length ( $l$ ) of each word.



**Figure 3.** Storage system concept of the memory unit.

### 3.2. Bi-LSTM Encoding with Attention

Having solved the exploding and vanishing gradient problems of RNN, LSTM has enjoyed a lengthy time-line of being the basic model widely used in NLP tasks due to its capability to capture long-term textual dependencies [19]. Moreover, the enhanced bi-directional variant of the traditional LSTM which comprises of 2 hidden layers in opposing directions enables it to connect past and future input features to the same output in specified time steps, thus, the Bi-LSTM is able to learn fast. Meanwhile, comparing Bi-LSTM to the bi-directional gated recurrent unit (Bi-GRU) [20,21], the study [20] showed that the latter is equally fast; however, we justify our choice of adopting the Bi-LSTM in the mDNA model based on the analysis in [22] that the LSTM is a better option than GRU for performing tasks where deep context understanding is the major consideration of our study.

In this section, we apply the two input sequences of memory and response,  $M$  and  $R$ , into the Bi-LSTM encoder to represent the tokens of the contextual vectors, and we denote the vectors of the hidden state as  $m^s$  and  $r^s$ :

$$m_i^s = \text{BiLSTM}(M, i), \quad (1)$$

$$r_j^s = \text{BiLSTM}(R, j), \quad (2)$$

where  $i$  and  $j$  represent the  $i$ -th context in the utterance, and the  $j$ -th context in the response, respectively.

To determine whether a selected response candidate is appropriate and correlated with the context of the utterance, modeling the relationship between the utterance and the response is an essential step. For example, appropriate responses consider contextual keyword vectors, which can be obtained by modeling the semantic relationship. To this regard, being that not all words contribute equally to the context representation of an utterance, attention is required to ensure that important information are collected for matching composition. In this case, we use a multi-head attention mechanism to align the utterance contexts to the response candidates, and then calculate the semantic relationship at the utterance level. We also applied a soft alignment layer to calculate the attention weights as follows:

$$e_{ij} = (m_i^s)^T * r_j^s, \quad (3)$$

where  $e \in R^{w*n}$ , and  $T$  is the length of the sequence. For the hidden state of utterance context, i.e.,  $m_i^s$  (already encoding the utterance and its contextual meaning), the relevant semantics in the response options are identified and composed using  $e_{ij}$  as a vector  $m_i^d$ , more specifically as shown in Equation (4).

$$\alpha_{ij} = \exp(e_{ij}) / \sum_{k=1}^n \exp(e_{ik}), m_i^d = \sum_{j=1}^n \alpha_{ij} r_j^s, \quad (4)$$

$$\beta_{ij} = \exp(e_{ij}) / \sum_{k=1}^w \exp(e_{kj}), r_j^d = \sum_{i=1}^w \beta_{ij} m_i^s, \quad (5)$$

where  $\alpha \in R^{w * n}$  and  $\beta \in R^{w * n}$  are the normalized attention weight matrices. The same process is also performed for each utterance and response matching in Equation (5).

Hence, we model the utterance level semantic relationship between the aligned utterance pairs, i.e.,  $\langle m_i^s, m_i^d \rangle$  and  $\langle r_i^s, r_i^d \rangle$ . The equation is further expressed as follows:

$$m_i^w = P([m_i^s; m_i^d; m_i^s - v_i^d; m_i^s \odot m_i^d]), \quad (6)$$

$$r_i^w = P([r_i^s; r_i^d; r_i^s - r_i^d; r_i^s \odot r_i^d]), \quad (7)$$

where a matching layer can be used to model some high-order interaction between the vectors  $m_i^w$  and  $r_i^w$  for the utterance and response, respectively.  $P$  is a 1-layer multi-layer perception (MLP) with a rectified linear unit (ReLU) activation.

### 3.3. Matching Composition and Prediction

In the process to predict the final response candidate selection, the Bi-LSTM is used to compose the matching vectors in a dense layer as follows:

$$m_i^m = \text{BiLSTM}(m_i^w, i), \quad (8)$$

and

$$r_j^m = \text{BiLSTM}(r_i^w, j). \quad (9)$$

Finally, we get the representations of DAM and mDNA denoted as  $[f_1, f_2]$  to compute the final matching score  $g(m, r)$ , which is formulated as:

$$g(m, r) = \text{softmax}(W_2[f_1, f_2] + b_2), \quad (10)$$

where  $W_2$  and  $b_2$  are the learning parameters. The loss function of our model is the negative log likelihood, defined as:

$$L = - \sum_D [y \log(g(m, r)) + (1 - y)(1 - \log(g(m, r)))], \quad (11)$$

where  $D$  is the dataset, and  $y$  is the label marked in  $D$ .

## 4. Experiments

### 4.1. Dataset

The availability and use of a large amount of training data promises an increase in the probability of achieving good performance in neural machine translation tasks. For this reason, we used the Ubuntu Dialogue Corpus [23] to train and analyse the performance of our mDNA model. The Ubuntu Corpus is an English dataset which contains multi-turn dialogues constructed from Ubuntu Internet Relay Chat (IRC) logs. The dataset consists of one million utterance-response pairs, and each pair have a binary label to mark the responses as either positive or negative (as shown in Table 1). In the training set, the ratio of the positive and the negative is 1:1, and 1:9 in the validation and test sets. Table 2 gives the statistics of the training set, validation set, and test set.

**Table 1.** Sample of Ubuntu Dialogue output with Matching Score of mDNA.

		Label	Matching Score
<b>Context</b>	A: Hi, I am looking to see what packages are installed on my system, I don't see a path in the list being held somewhere else. B: try dpkg – get-selections. A: Is that like a database for packages instead of a flat file structure? B: dpkg is the debian package manager-get-selections that simply shows you what packages are handled by it.	1	0.995
<b>Response</b>	No clue what do you need it for, its just reassurance as I don't know the debian package manager	1	0.995
	Then why not +q good point thanks.	0	0.231
	I mean real media ... not the command lol exactly.	0	0.035
	Hmm: should i force version to hoary.	0	0.299
	I will also run on core2 intels i installed ubuntu on a usb ...	0	0.016
	And what's your system specs the live cd does not use the hard drive at all	0	0.019
	Thanks i will see if i can find another option and use that as a last resort the steps ...	0	0.911
	Number is long term support also and not that much difference between number and number with the next ubuntu kubuntu and xubuntu at the end of april.	0	0.02
	These days backup to usb probably is more relevant ...	0	0.004
	It explains the same stuff a bit more in depth:) ...	0	0.963

**Table 2.** Statistics of Ubuntu Corpus dataset.

	Training	Validation	Testing
# context response pairs	1 M	500 K	500 K
# candidates per context	2	10	10
Avg. # turns per dialogue	7.71	7.33	7.54
Avg. # words per utterance	10.34	10.22	10.33

#### 4.2. Hyperparameter Settings

We implemented the mDNA with Python 3 language, and used word2vec [24] to pre-train the word embedding on training set. The hidden size of the Bi-LSTM layers is set to 300, and to further enable fine-grain deep learning for good performance, the number of training epoch is set to 15 which we arrived at by fine-tuning the model with early stopping mechanism, while we set the mini-batch sizes to 16 for training, testing and validation, respectively, as could be accommodated by the Titan NVIDIA GPU memory resource available in our machine. Furthermore, the initial learning rate is set as  $2 \times 10^{-4}$  (i.e., 0.0002) used to update the learning parameters by stochastic gradient descent with Adam optimizer [25]. While low learning rate tends to slow down the training speed, it also contributes to steep decrease in the network's losses. Finally, the mDNA model is trained in Tensorflow-GPU environment with Cuda kernel, and the learning parameters  $W_2$  and  $b_2$  in the softmax function uses the default value set by Tensorflow. Being that Ubuntu dialog corpus is large (i.e., comprising of 1,000,000 Q&A pairs), we set the vocabulary size to 100,000; and lastly, we assigned the maximum lengths of utterance input embedding and response input embedding as 350 and 150, respectively.

#### 5. Results and Evaluation

In this paper, we used the recall R at position  $k$  in  $n$  number of candidates, denoted as  $R_n@k$ , as the evaluation metrics, as it is also used in DAM [1], SMN [2], and other novel retrieval chatbot models [20,24,26–29] we selected to compare their performance with that of the mDNA model. Here,  $R_n@k$  is defined as (the number of relevant retrieved response candidates at top- $k$ )/(total  $n$  number of retrieved response candidates). We represent this mathematically as below:

$$R_n@k = \frac{TP}{TP + FN}, \quad (12)$$

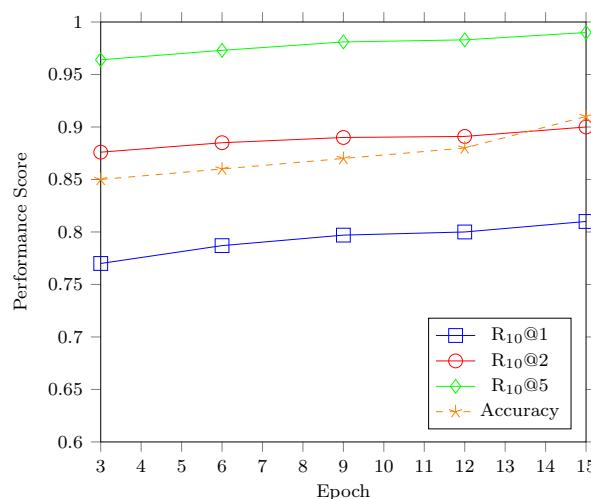
where TP and FN represent true positive and false negative of retrieved response candidates, respectively. This evaluation metrics formula is suitable for information retrieval tasks, and is popularly used to measure and score the performance of information retrieval systems in terms of the degree at which the selected output response candidates match with the context of the input utterances. In our task, the mDNA model needs to select a response from  $n$  number of candidate responses. In this process, if the true response is among the response candidates, it is referred to as the true positive. Figure 4 shows the overview representation plot of  $R_{10}@1, 2$ , and 5, with respect to different values of epochs which shows that the model achieved a stable training (and learning). Moreover, to further analyze the performance of our model, we selected some representative models based on retrieval dialog systems (that are either based on Transformer or gated-RNN respective), DAM [1], SMN [2], DualEncoder [23], MV-LSTM [26], Match-LSTM [27], Multiview [28], DL2R [29], and BERT Bi-Encoder+CE [20]. We compared the performance of these models to that of the mDNA model; and to perform ablation analysis, we removed the DAM model combined with the mDNA, and referred to it as  $mDNA_{self}$ .

In Table 3, the recall evaluation results show that the mDNA performs competitively better than most of the selected chatbot models trained on Ubuntu Dialogue Corpus that were compared to it. With respect to the baseline DAM model, the performance of the mDNA increased by 3% on

$R_{10}@1$ , 1.7% on  $R_{10}@2$  and 0.6% on  $R_{10}@5$ , respectively. While the mDNA outperforms all other selected models compared to it, we observe that the recall evaluation scores of the mDNA and BERT Bi-Encoder+CE model [20] are similar, and therefore the performance gap between them is very insignificant. The BERT Bi-Encoder+CE model exploits the benefits of Bi-GRU and Bi-encoder system of BERT-Transformer architecture which enables its performance to be comparable to that of the mDNA. However, the capability of augmenting Transformer-based retrieval chatbot models for performance enhancement is a unique attribute to which the mDNA claims superiority.

**Table 3.** Performance evaluation results of the mDNA and some selected retrieval chatbot models trained on Ubuntu dataset.

Model	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
DualEncoder [23]	0.638	0.784	0.949
MV-LSTM [26]	0.653	0.804	0.946
Match-LSTM [27]	0.653	0.799	0.944
Multiview [28]	0.662	0.801	0.951
DL2R [29]	0.626	0.783	0.944
SMN [2]	0.726	0.847	0.961
DAM [1]	0.767	0.874	0.969
BERT Bi-Encoder + CE [20]	0.793	0.893	0.975
<b>mDNA</b>	<b>0.797</b>	<b>0.891</b>	<b>0.975</b>
<b><math>mDNA_{self}</math></b>	<b>0.788</b>	<b>0.885</b>	<b>0.971</b>



**Figure 4.** Plot showing increase in average accuracy (denoted as accuracy) scores and performance evaluation scores as the number of training epoch increases. The accuracy is averaged by  $R_{10}@1$ ,  $R_{10}@2$ , and  $R_{10}@5$  at corresponding epochs.

Lastly, owing to the important role of memory in the mDNA, we investigated to see if adjustment of the maximum length of utterance embedding is fed into the memory would affect the performance and effectiveness in any way. In this task, we initially set the dimension of utterance memory embedding to 350 and, subsequently, during multiple retraining sessions of the network, we altered the value from 350 to 300, 250, 200, and 150 at different corresponding intervals of the following training epochs: 3, 6, 9, 12, and 15, respectively. After several retraining of the model, we observed that the size of utterance word embedding parsed into the memory does not affect its performance, but instead, altering the values of the epochs directly influences overall accuracy and performance of the model. As the number of epoch increases from a lower value to the max set value, the average accuracy score also increases, as well as the scores of  $Recall_{10}@1$ ,  $Recall_{10}@2$ ,  $Recall_{10}@5$ , and vice-versa. This scenario is depicted with the plot in Figure 4.

## 6. Conclusions

In this paper, we presented the memory-based deep neural attention (mDNA) method and described the concepts of its operational process as well as the benefits of using fusion technique to augment independent retrieval chatbot models that are based on the Transformer architecture, for enhancing response selection accuracy, and consistency in logical context-dependency matching, respectively. As a case study, we selected the DAM network model and hybridised it with the mDNA to demonstrate the effectiveness of achieving better performance. The results obtained show the improvement in performance over some existing state-of-the-art retrieval chatbots. In future works, we will consider enhancing the Bi-LSTM encoder algorithm to support multimodal input-data and fusion (such as text and image) which will necessitate the use of more diversified datasets in the training of the model to further improve the accuracy of response selection.

**Author Contributions:** Conceptualization, J.C.; methodology, O.A.; writing—original draft preparation, J.C.; software, O.A.; validation, J.C. and O.A.; visualization, J.C.; supervision, L.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2221-E-182-042-MY2; and in part by the Chang Gung Memorial Hospital, Kweishan, Taoyuan, Taiwan, under Grants CMRPD2J0012 and CMRPD2I0052.

**Conflicts of Interest:** The authors declare no conflict of interests regarding the publication of this article.

## References

1. Zhou, X.; Li, L.; Dong, D.; Liu, Y.; Chen, Y.; Zhao, W.X.; Yu, D.; Wu, H. Multi-Turn Response Selection for Chatbots with Deep Attention Matching Network. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 1118–1127.
2. Wu, Y.; Wu, W.; Xing, C.; Li, Z.; Zhou, M. Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-based Chatbots. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 496–505.
3. Leuski, A.; Traum, D. NPCEditor: Creating Virtual Human Dialogue Using Information Retrieval Techniques. *AAAI AI Mag.* **2011**, *32*, 42–56. [[CrossRef](#)]
4. Wang, H.; Lu, Z.; Li, H.; Chen, E. A Dataset for Research on Short-Text Conversation. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 935–945.
5. Hu, B.; Lu, Z.; Li, H.; Chen, Q. Convolutional neural network architectures for matching natural language sentences. In Proceedings of the 27th Conference Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2042–2050.
6. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Conference Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
7. Medina, J.R.; Kalita, J. Parallel Attention Mechanisms in Neural Machine Translation. In Proceedings of the 17th IEEE International Conference on Machine Learning and Applications, Orlando, FL, USA, 17–20 December 2018; pp. 547–552.
8. Zhang, Z.; Li, J.; Zhu, P.; Zhao, H.; Liu, G. Modeling Multi-turn Conversation with Deep Utterance Aggregation. In Proceedings of the 27th International Conference Computational Linguistics, Santa Fe, NM, USA, 21–25 November 2018; pp. 3740–3752.
9. Liu, K.; Li, Y.; Xu, N.; Natarajan, P. Learn to combine modalities in multimodal deep learning. *arXiv* **2018**, arXiv:1805.11730.
10. Henderson, M.; Vulic, I.; Gerz, D.; Casanueva, I.; Budzianowski, P.; Coope, S.; Spithourakis, G.; Wen, T.; Mrkšić, N.; Su, P. Training Neural Response Selection for Task-Oriented Dialogue Systems. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5392–5406.

11. Zhao, X.; Tao, C.; Wu, W.; Xu, C.; Zhao, D.; Yan, R. A Document-grounded Matching Network for Response Selection in Retrieval-based Chatbots. In Proceedings of the 28th IJCAI Conference on AI, Macao, China, 10–16 August 2019; pp. 5443–5449.
12. Yang, L.; Ai, Q.; Guo, J.; Croft, W.B. aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model. In Proceedings of the 25th ACM International Conference Information and Knowledge, Indianapolis, IN, USA, 24–28 October 2016; Volume 8, pp. 287–296.
13. Lakew, S.M.; Cettolo, M.; Federico, M. A Comparison of Transformer and Recurrent Neural Networks on Multilingual Neural Machine Translation. In Proceedings of the 27th International Conference Computational Linguistics, Santa Fe, NM, USA, 20–25 August 2018; pp. 641–652.
14. Wang, Z.; Ma, Y.; Liu, Z.; Tang, J. R-transformer: Recurrent Neural Network Enhanced Transformer. *arXiv* **2019**, arXiv:1907.05572.
15. Sukhbaatar, S.; Szlam, A.; Weston, J.; Fergus, R. End-to-end Memory Networks. In Proceedings of the Conference on Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 1, pp. 2440–2448.
16. Kaiser, Ł.; Bengio, S. Can Active Memory Replace Attention? In Proceedings of the 30th Conference Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; pp. 3781–3789.
17. Wulamu, A.; Sun, Z.; Xie, Y.; Xu, C.; Yang, A. An Improved End-to-End Memory Network for QA Tasks. *Cmc Comput. Mater. Contin.* **2019**, *60*, 1283–1295. [[CrossRef](#)]
18. Domhan, T. How Much Attention Do You Need? A Granular Analysis of Neural Machine Translation Architectures. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 1799–1808.
19. Zhou, Q.; Wu, H. NLP at IEST 2018: BiLSTM-Attention and LSTM-Attention via Soft Voting in Emotion Classification. In Proceedings of the 9th ACM Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Brussels, Belgium, 31 October 2018; pp. 189–194.
20. Vakili, A.; Shakery, A. Enriching Conversation Context in Retrieval-based Chatbots. *arXiv* **2019**, arXiv:1911.02290v1.
21. Chen, J.; Abdul, A. A Session-based Customer Preference Learning Method by Using the Gated Recurrent Units with Attention Function. *IEEE Access* **2019**, *7*, 17750–17759. [[CrossRef](#)]
22. Gruber, N.; Jockisch, A. Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text? *J. Front. Artif. Intell.* **2020**, *3*, 1–6. [[CrossRef](#)]
23. Lowe, R.; Pow, N.; Serban, I.; Pineau, J. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Prague, Czech Republic, 2–4 September 2015; pp. 285–294.
24. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 27th Annual Conference Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
25. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference Learning Representations, San Diego, CA, USA, 7–9 May 2015; Volume 1412.
26. Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Wan, S.; Cheng, X. Text matching as image recognition. In Proceedings of the 30th AAAI Conference Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 2793–2799.
27. Wang, S.; Jiang, J. Machine Comprehension using Match-LSTM and Answer Pointer. In Proceedings of the International Conference Learning Representations, Palais des Congrès Neptune, Toulon, France, 24–26 April 2017; pp. 2793–2799.
28. Zhou, X.; Dong, D.; Wu, H.; Zhao, S.; Yu, D.; Tian, H.; Liu, X.; Yan, R. Multi-view response selection for human-computer conversation. In Proceedings of the EMNLP 2016, Austin, TX, USA, 1–5 November 2016; pp. 372–381.
29. Yan, R.; Song, Y.; Wu, H. Learning to respond with deep neural networks for retrievalbased human-computer conversation system. In Proceedings of the SIGIR 2016, Pisa, Italy, 17–21 July 2016; pp. 55–64.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/368759521>

# Corpus Pragmatics

Book · February 2023

DOI: 10.1017/9781009091107

---

CITATIONS

0

READS

81

4 authors, including:



Daniela Landert

Heidelberg University

28 PUBLICATIONS 155 CITATIONS

[SEE PROFILE](#)



Daria Dayter

Tampere University

37 PUBLICATIONS 326 CITATIONS

[SEE PROFILE](#)



Thomas C. Messerli

University of Basel

29 PUBLICATIONS 127 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Forschungslogiken in the text-based Digital Humanities: Analysing evaluative practices after the machine learning turn [View project](#)



FUNGRESSION: Humour and impoliteness on social media [View project](#)



**Cambridge  
Elements**

Pragmatics

# Corpus Pragmatics

Daniela Landert,  
Daria Dayter,  
Thomas C. Messerli  
and Miriam A. Locher



# Cambridge Elements

Elements in Pragmatics

edited by

Jonathan Culpeper

*Lancaster University, UK*

Michael Haugh

*University of Queensland, Australia*

## CORPUS PRAGMATICS

Daniela Landert

*Heidelberg University*

Daria Dayter

*Tampere University*

Thomas C. Messerli

*University of Basel*

Miriam A. Locher

*University of Basel*



CAMBRIDGE  
UNIVERSITY PRESS



Shaftesbury Road, Cambridge CB2 8EA, United Kingdom  
One Liberty Plaza, 20th Floor, New York, NY 10006, USA  
477 Williamstown Road, Port Melbourne, VIC 3207, Australia  
314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025,  
India  
103 Penang Road, #05-06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of Cambridge University Press & Assessment, a department of the University of Cambridge.

We share the University's mission to contribute to society through the pursuit of education, learning and research at the highest international levels of excellence.

[www.cambridge.org](http://www.cambridge.org)

Information on this title: [www.cambridge.org/9781009095082](http://www.cambridge.org/9781009095082)

DOI: [10.1017/9781009091107](https://doi.org/10.1017/9781009091107)

© Daniela Landert, Daria Dayter, Thomas C. Messerli and Miriam A. Locher 2023

This work is in copyright. It is subject to statutory exceptions and to the provisions of relevant licensing agreements; with the exception of the Creative Commons version the link for which is provided below, no reproduction of any part of this work may take place without the written permission of Cambridge University Press.

An online version of this work is published at [doi.org/10.1017/9781009091107](https://doi.org/10.1017/9781009091107) under a Creative Commons Open Access license CC-BY-NC-ND 4.0 which permits re-use, distribution and reproduction in any medium for non-commercial purposes providing appropriate credit to the original work is given. You may not distribute derivative works without permission. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-nd/4.0>

All versions of this work may contain content reproduced under license from third parties.

Permission to reproduce this third-party content must be obtained from these third-parties directly.

When citing this work, please include a reference to the DOI [10.1017/9781009091107](https://doi.org/10.1017/9781009091107)

First published 2023

*A catalogue record for this publication is available from the British Library.*

ISBN 978-1-009-09508-2 Paperback

ISSN 2633-6464 (online)

ISSN 2633-6456 (print)

Cambridge University Press & Assessment has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

# Corpus Pragmatics

## Elements in Pragmatics

DOI: 10.1017/9781009091107  
First published online: February 2023

Daniela Landert  
*Heidelberg University*

Daria Dayter  
*Tampere University*

Thomas C. Messerli  
*University of Basel*

Miriam A. Locher  
*University of Basel*

**Author for correspondence:** Daniela Landert,  
[daniela.landert@as.uni-heidelberg.de](mailto:daniela.landert@as.uni-heidelberg.de)

**Abstract:** This Element discusses the challenges and opportunities that different types of corpora offer for the study of pragmatic phenomena.

The focus lies on a hands-on approach to methods and data that provides orientation for methodological decisions. In addition, the Element identifies areas in which new methodological developments are needed in order to make new types of data accessible for pragmatic research. Linguistic corpora are currently undergoing diversification. While one trend is to move towards increasingly large corpora, another trend is to enhance corpora with more specialised and layered annotation. Both these trends offer new challenges and opportunities for the study of pragmatics. This Element provides a practical overview of state-of-the-art corpus pragmatic methods in relation to different types of corpus data, covering established methods as well as innovative approaches. This title is also available as Open Access on Cambridge Core.

**Keywords:** corpus-assisted pragmatics, corpus-driven pragmatics, multimodality, function-to-form mapping, comparability of corpus data

© Daniela Landert, Daria Dayter, Thomas C. Messerli and Miriam A. Locher 2023

ISBNs: 9781009095082 (PB), 9781009091107 (OC)  
ISSNs: 2633-6464 (online), 2633-6456 (print)

# Contents

1	Introduction	1
2	Corpora and Their Characteristics: Challenges and Opportunities for Pragmatic Research	9
3	Corpus-Driven Approaches with Self-Compiled Corpora	20
4	Corpus-Assisted Approaches with Self-Compiled Corpora	27
5	Compatibility and Comparability: Combining Existing Corpora	39
6	Scalability: Meaningful Pragmatics with Large Data	48
7	Multimodality: Integrating Non-verbal Information	56
8	Open Issues and Outlook	63
	Appendix A: Corpora	68
	Appendix B: Corpus Tools and Additional Resources	72
	References	73

## 1 Introduction

### 1.1 Corpora and Pragmatics

The role of corpora in pragmatic research has continually gained importance over time, and the number of published handbooks (Aijmer & Rühlemann 2014), text books (Rühlemann 2019), edited volumes (e.g. Romero-Trillo 2008; Jucker, Schreier & Hundt 2009; Suhr & Taavitsainen 2012; Taavitsainen, Jucker & Tuominen 2014) and the launch of a journal (*Corpus Pragmatics*, since 2017) demonstrate clearly that we have moved far beyond the point at which it has to be argued why corpus approaches are suitable and valuable for pragmatic studies. Corpus approaches have gained their place next to introspection, experimentation and non-corpus-based observational approaches (see Jucker, Schneider & Bublitz 2018). At the same time, there still remain many challenges for corpus approaches to pragmatic research questions. For instance, corpora still tend to be characterised by a lack of access to context, privileging of quantitative results over qualitative interpretation and focus on linguistic forms rather than their functions, all of which can hinder pragmatic studies. We believe that by taking a closer look at these challenges, we can identify new avenues in which to develop corpus pragmatics.

This introductory section presents a brief overview of the main challenges with which corpus pragmatics is faced. These challenges provide the starting point for the following sections, which deal with ways to overcome some of these challenges as well as highlighting the many benefits corpora offer for pragmatics. In doing so, we focus on the wide variety of different corpora that have become available to researchers and we explore the opportunities that each of these types of corpora provides for pragmatic studies and the research avenues that can be pursued. We adopt a wide understanding of corpora, which includes not only prototypical large and publicly released corpora, but also small, purpose-built, ad hoc collections of data. Likewise, we discuss a broad range of different approaches. What they all have in common is that they rely on digital data compilations and that they aim to identify, systematically search for and analyse linguistic patterns and their pragmatic functions with the help of computers. We place special emphasis on function-to-form approaches to corpus pragmatics. This is not to imply that form-to-function approaches are not suitable or relevant for pragmatics – they certainly are, as has been amply demonstrated. Our focus on function-to-form approaches is due to the unique challenge that they pose, given that they do not start with form-based search patterns (see Section 1.4). Since all authors of this Element work in the field of English linguistics, we will focus our discussion on English language corpora. However, the number and variety of corpora is large and continuously growing for many other languages as well, and the methods we discuss can

equally be applied to corpora of other languages. Due to space restrictions, we cannot present an extensive overview of the research field, but we refer to relevant resources where interested readers can find more detailed information and overviews of the field.

## 1.2 Context

Pragmatic meaning is influenced by all levels of context (see [Garcés-Conejos Blitvich & Sifianou 2019](#)). On the micro level, the immediate co-text of an expression influences its interpretation. On the meso level, pragmatic meaning depends on text types, genres and thematic domains. And on the macro level, the social, cultural and historical context affects pragmatic meaning as well. As a consequence, access to contextual information is often crucial for pragmatic studies. This reliance on access to context presents what is perhaps the biggest challenge for corpus pragmatics overall. While there are some corpora that provide excellent support for accessing contextual information, many corpora – especially those that were created for studying grammar – do not sufficiently satisfy this requirement.

At the level of micro-context, access to the co-text is restricted by many corpus interfaces. Search tools mostly present results in concordance views that show the search term with a minimal co-text of a number of words to the left and right. While this word span is usually sufficient for disambiguating word meaning, it often does not provide enough information for a pragmatic analysis. Even corpus interfaces that support an expanded context view, such as the *Corpus of Contemporary American English (COCA)* and other corpora by the Davies corpus family, often restrict the expanded context to no more than a few lines (information on all corpora mentioned in this Element can be found in [Appendix A](#)). For some pragmatic interpretations, a few lines of context may be sufficient, but sometimes access to an entire text or conversation is crucial in order to understand how a specific utterance can be interpreted, or whether several interpretations are possible. Other popular corpus tools, like keyword analysis, collocation analysis and n-gram analysis, present results that are even more decontextualised than concordance lines.

There are at least two reasons for the tendency to restrict access to the co-text in corpora. First, copyright restrictions often keep corpus compilers from sharing full texts with users. This is the case, for instance, with many corpora that are based on films and TV series, such as the *Sydney Corpus of Television Dialogue (SydTV)*, as well as the *SOAP Corpus*, the *Movie Corpus* and the *TV Corpus* from the Davies corpus family ([Bednarek et al. 2021: 2–3](#)). Second, most corpora that are released to the research community were not built

primarily for studying pragmatics. Instead, corpus compilers often focus on grammatical and lexical characteristics. The focus of analysis influences sampling procedures and the development of corpus tools. For instance, many balanced reference corpora include text samples of a limited size, rather than entire texts. For studies of grammar, this has advantages, since it limits the possibility of the corpus being biased due to overrepresentation of individuals. For studies of pragmatics, this is true as well, but at the same time the lack of access to the larger co-text creates problems for the analysis.

With respect to the meso context, i.e. information about the genre or text type, and the thematic domain from which the corpus data was sampled, the information provided by corpora varies a great deal. Many corpora are built around one or more classifications that are relevant for studying variation across genres or text types. While some general information about this categorisation is often available, more fine-grade and more detailed information may be more difficult to access. For instance, while many balanced reference corpora include a category of news texts, distinctions into hard news dealing with political topics and soft news dealing with human interest content are less common. However, this distinction can greatly affect the use of expressions and, thus, such information can be relevant for pragmatic studies. Meso-level classifications can become especially challenging in comparisons across different corpora, when the same label is applied but the data sets are compiled according to different principles. The creation of suitable classifications that can be generalised across different corpora and types of data is often not possible, and classifications tend to vary between corpus compilers. The documentation of the sampling principles can provide valuable information on the kind of data that is included in the corpus, but sometimes this information is not easily available or not very detailed (see also [Section 5](#)).

When it comes to the macro context, the situation varies greatly across different types of corpora. Specialised corpora that include data from one specific domain are often published together with studies that present extensive discussions of the kind of data included in the corpus. In contrast, balanced reference corpora like the *British National Corpus* (*BNC*) and the *COCA* tend to assume that users are familiar with the variety that is represented. The challenge of lack of access to information about the macro context is perhaps greatest for historical corpora, where familiarity with the general socio-historical context can be taken for granted at the least.

How marked the effect of research focus can be on the composition and metainformation of a corpus becomes apparent if we look at corpora that were composed with pragmatic research questions in mind. A case in point are pragmatically annotated corpora, where pragmatic information is included

during the composition process (see [Section 2.6](#)). A different example can be found in certain historical corpora which were composed with a wide range of possible research aims in mind, including pragmatic questions. An example of this is the three-part *Corpus of Early English Medical Writing 1375–1800* (*CEEM*). The corpus was compiled as part of a long-running research project investigating changes in scientific thought-styles, hosted at the Research Unit for Variation, Contacts and Change in English at the University of Helsinki. The pragmatic nature of the research interest affected the corpus composition in a number of ways: the corpus includes long, 10,000 word extracts of texts rather than short samples; it comes with its own software, which provides access to passages in the context of the entire extract; it includes extensive metainformation on each text and author; and it provides direct links to additional context information, such as hyperlinks to scanned manuscript pages on *Early English Books Online* (*EEBO*) (see [Tyrrkkö, Hickey & Marttila 2010](#)). All of this makes the corpus very well-suited for studies of pragmatic variables, while it can nevertheless be fruitfully used to investigate purely grammatical or lexicological research questions (e.g. [Méndez-Naya & Pahta 2010](#)).

### 1.3 Qualitative and Quantitative Analysis

Due to the various levels of context-dependency of pragmatic meaning, pragmatic analysis often relies quite strongly on interpretation and qualitative analysis. Researchers make judgements about the pragmatic functions of utterances by interpreting them in the context in which they are used through what is sometimes referred to as horizontal reading ([Rühlemann & Aijmer 2014](#): 3), followed by subsequent classification. This process is time-consuming and not easily reconciled with the most prototypical uses of standard corpus tools. Most corpora are developed with the aim to support quantitative analysis and so-called vertical reading of data. They offer tools such as keyword in context (KWIC) views, normalised frequencies, keywords, collocations and n-grams. All of these tools present decontextualised or only partially contextualised views of results that help establish broader patterns across the entire corpus. When these tools are used to carry out pragmatic analysis, researchers often need to find ways of combining them with more contextualised perspectives.

In this Element, we discuss different ways in which this tension between qualitative and quantitative approaches can be resolved. The focus of [Section 4](#) lies on corpus-assisted approaches that rely very heavily on qualitative analysis and, as a consequence, use small corpora, which still make it possible to quantify patterns in the data. In [Sections 6](#) and [7](#), we focus on two areas in which we see a demand for the development of new corpus methods that take

pragmatic research questions into consideration. For [Section 6](#), these are methods that help researchers apply qualitative methods to large corpora in meaningful ways, and [Section 7](#) turns to the analysis of multimodal data, where there is a great deal of demand for new methods that support pragmatic analysis, especially when it comes to quantifying observations.

## 1.4 Form-to-Function and Function-to-Form Mapping

Most pragmatic research deals with the relationship between linguistic forms and their functions. This relationship can be studied in two different ways. In form-to-function approaches, researchers study the different pragmatic functions of a given form, whereas in function-to-form approaches the aim is to identify forms with which a given function can be expressed ([Jacobs & Jucker 1995](#): 13). Corpus pragmatics is particularly well-suited for form-to-function approaches, since corpora make it possible for researchers to search for all surface forms within the data ([O'Keeffe 2018](#): 587). As [O'Keeffe et al. \(2020](#): 47) note, corpus linguistics overall is form-to-function oriented: apart from special cases, such as pragmatically annotated corpora, corpus searches always start with forms. Thus, it is probably no surprise that much of the early research in corpus pragmatics adopted form-to-function approaches. The study of pragmatic markers (or discourse markers) proved especially productive. An influential example of such studies is [Aijmer's \(2002\)](#) research monograph, which looked at a range of discourse particles, including *now*, *oh* and *ah*, and *actually* in the *London-Lund Corpus*. Other corpus-based studies on pragmatic markers have been carried out on a wide range of Present-day English (e.g. [Aijmer 2008](#); [Buyssse 2012](#); [Kirk 2015](#); [Beeching 2016](#)) and historical corpora (e.g. [Culpeper & Kytö 2010](#): ch. 15; [Lutzky 2012](#)). Other examples of form-to-function approaches include the study of stance markers like *I think* (e.g. [Aijmer 1997](#); [Kärkkäinen 2003](#); [Simon-Vandenbergen 2000](#)) and the earlier form *methinks* (e.g. [Palander-Collin 1999](#)); the study of the pragmatic functions of *basically* ([Butler 2008](#)); the study of interjections (e.g. [Norrick 2009](#)) and the study of the planners *uh* and *um* (e.g. [Tottie 2011, 2014, 2019](#); [Tonetti Tübben & Landert 2022](#)).

In contrast, function-to-form approaches are faced with the challenge that functions cannot be searched for automatically. This has not deterred researchers from investigating pragmatic functions in corpora, though. Speech acts in particular have been investigated with function-to-form approaches. An early example is [Kohnen's \(2000\)](#) study of directive speech acts in the *Lancaster–Oslo/Bergen Corpus* and the *London-Lund Corpus*. The difficulty of overcoming the lack of one-to-one correspondence between

form and function is discussed very prominently in this paper. In his conclusion, [Kohnen \(2000: 184\)](#) states: ‘It seems probable that the fundamental difficulty, the open relationship between form and function, cannot be solved . . . Perhaps a corpus-based study of speech acts will have to focus on the patterns representing the most typical and common manifestations of a speech act and will not seek to cover all the possible manifestations of that speech act.’ It would be overly optimistic to claim that these problems have been completely resolved in the meantime. However, research on speech acts and other function-to-form topics has made much progress (for an overview of types of approaches, see [Jucker 2013](#)). Approaches that have been developed include the search for illocutionary force indicating devices (IFIDs) (e.g. [Deutschmann 2003](#); [Jucker & Taavitsainen 2008](#)), the search for known lexico-grammatical patterns (e.g. [Adolphs 2008](#); [Jucker et al. 2008](#)) and the study of metacommunicative expressions (e.g. [Jucker & Taavitsainen 2014](#); see also [Haugh 2018](#)). For research of stance expressions, the analysis of high-density passages that include clusters of known expressions has been shown to be an effective method, which may have potential for other research areas as well ([Landert 2019](#)).

Since automatic retrieval always takes place based on surface forms, even function-to-form approaches usually rely on the identification of forms in the corpus data. Exceptions to this can be found in pragmatically annotated corpora and in small-scale corpus-based approaches. In both cases, the corpus data is read and interpreted by a researcher, either as part of the compilation process (annotated corpora) or as the main focus of the analysis (small-scale corpus-based approaches). It remains an open question to what extent the identification of pragmatic functions will ever be possible without relying on predefined forms and manual evaluation.

### [1.5 Scalability](#)

Scalability refers to the ease with which research methods can be applied to large sets of data. If a method is scalable, then processing a very large amount of data does not take considerably longer than processing a small amount of data. Generally speaking, quantitative methods have a much better scalability than qualitative methods. For instance, if we are interested in comparing normalised frequencies across different sets of data, the size of the data only marginally affects the duration of the calculation, as long as the number of hits can be retrieved automatically. While the automatic computation of the number of hits in the corpus increases with corpus size, it still is so fast that the duration of computation takes up a very small proportion of the overall

research time. In contrast, the manual classification of pragmatic functions increases with the size of data in a linear fashion. Classifying 10,000 instances takes, roughly, 100 times as long as classifying 100 instances. In other words, manual classifications have very poor scalability.

Compared to the other challenges discussed in this section, scalability has received little attention so far. However, scalability is at the core of all of the previously discussed challenges. It is due to the poor scalability of context-dependent interpretations and the manual work involved in identifying pragmatic functions that corpora present challenges for pragmatic studies. This issue is becoming more pressing with the trend towards increasing corpus size. Larger corpora provide more data and, as a consequence, potentially more opportunities for insightful observations into pragmatic phenomena, but the large amount of work involved in qualitative analysis means that either only a small section of the data can be considered or that the focus shifts even more towards quantitative evaluations. This does not have to be the case, though. While linguists have spent a great amount of effort on developing new quantitative methods of exploring corpus data, there has been far less exploration of the ways in which we can support the qualitative analysis of data in larger corpora. As we will discuss in [Section 6](#), it is possible, for instance, to develop semi-automated methods which rely on automated procedures to retrieve particularly relevant instances of a phenomenon for further manual analysis. Such methods help make qualitative analysis more scalable and thus open up new research perspectives for working with large corpora, even when the phenomenon under investigation relies on manual analysis.

## 1.6 Corpus Pragmatics and its Advantages

Despite all these challenges, corpus pragmatics is a vibrant field of research that can yield meaningful insights that complement studies on the same phenomena that were derived with different methodologies. It addresses a wide variety of research questions and makes use of a large range of corpus resources, tools and approaches. In this Element, we focus on this variety and point out those areas in which we see most potential for further developments. Throughout the Element, we highlight why it might be worth doing corpus pragmatics in the first place. Some of the advantages that corpus approaches offer can be summarised as follows:

- Pattern finding: Corpora present well-defined data sets that make it possible for researchers to identify reoccurring patterns.
- Systematicity: Corpora include samples that represent language use in a given variety or domain in a systematic way. This makes it possible to

assess the extent to which corpus data is representative of the variety or domain overall.

- Generalisation: Observations of reoccurring patterns across different sets of data make it possible to generalise findings for the language use that is represented in the corpus, provided that the corpus is compiled in a way that such generalisations are legitimate (e.g. with respect to representative sampling).
- Reproducibility: To the extent that corpus data is accessible to other researchers, and provided that methodological explanations are presented with a sufficient amount of detail, results that are based on corpora can be verified by other researchers.
- Transparency: By making use of often well-established corpus linguistic methodology, corpus pragmatics can present findings in such a way that the methodological steps that lead to them are maximally transparent.

## 1.7 Outline of the Element

In this Element, we do not attempt to present a comprehensive overview of corpus pragmatics. There are already various handbooks and textbooks that present excellent overviews of how different types of pragmatic phenomena can be studied with corpus data and methods (see [Section 1.1](#)). Instead, we focus on the challenges that we introduced in this section and on possible ways of addressing and overcoming them. We also want to emphasise that the term corpus can be used to refer to vastly different types of data collections: from small, manually curated compilations of project-specific datasets to huge (semi-)automatically compiled collections of electronically available documents; from data formatted and annotated for a given research objective to corpora that are compiled to ensure maximum compatibility with existing and future resources; and from purely text-based corpora to corpora including audio and video files and visual resources. This Element is structured along these distinctions, emphasising that each of these types of corpora presents unique challenges and opportunities for corpus pragmatic research.

[Section 2](#) gives an overview of different types of corpora and how their characteristics influence pragmatic research. Each type is illustrated with examples from corpora that have been used for pragmatic studies in the past. Appendices A and B complement this section by providing lists of all corpora ([Appendix A](#)) and all corpus tools ([Appendix B](#)) mentioned in the Element. [Sections 3](#) and [4](#) discuss two different ways in which purpose-built corpora can be used. In [Section 3](#), the focus lies on corpus-driven research on project-specific data. The section discusses the manifold benefits, as well as

the challenges of working with data compiled from online sources, including ethics, copyright issues and data documentation. While these issues of corpus composition are discussed in the context of corpus-driven research, many of them equally apply to corpus-based studies with self-compiled corpora. **Section 4** turns to much smaller purpose-built corpora and the ways in which they can be used for corpus-assisted discourse analysis. The case study presented in the section illustrates the analytic steps needed to apply qualitative methods in such a way that quantifiable patterns can be identified in the data. **Section 5** turns to the many practical challenges involved in combining different existing corpora within a single study. These challenges include, for instance, differences in data formats, annotation, metainformation and access to data, and are discussed under the keywords compatibility and comparability. While incompatibility of corpus formats can make it more difficult to work with different corpora side-by-side, the lack of comparability of data and results within or across corpora will negatively affect the reliability of results and interpretations. **Section 6** deals with the issue of scalability and argues for the value of developing new scalable methods that make it possible to use large corpora even for research questions that rely on qualitative data analysis. In **Section 7**, we present an overview of the most recent developments in the area of multimodal corpora and discuss what is needed for pragmatic studies of such corpora. **Section 8** concludes the Element with a summary of open issues and an outlook to possible future developments concerning new types of corpus resources, new methods and new research questions.

## 2 Corpora and their Characteristics: Challenges and Opportunities for Pragmatic Research

### 2.1 Introduction

In this Element, we adopt a broad perspective on corpus pragmatics and corpora. We present an overview of the many different ways in which corpora can be used for pragmatic research. In order to do so, it is necessary to point out the wide variety of linguistic corpora that exist today, which is the topic of this section. Our aim is not to present a comprehensive list of all corpora – not only would this be an impossible task but the list would be outdated very quickly – but rather to introduce distinctions between types of corpora as discussed in the literature, many of which will be relevant for the remaining sections in the Element.

In what follows, we decided to introduce examples of what the research community has coined at various times as types of corpora of interest to pragmatics: balanced reference corpora, topic- and domain-specific corpora,

spoken corpora, multimodal corpora, pragmatically annotated corpora, learner corpora, parallel multilingual corpora, corpora including unsystematic large text collections and purpose-built self-compiled corpora. However, the distinctions between these categories are in fact fuzzy and they are not mutually exclusive. In other words, corpora can belong to more than one group. For instance, there are balanced reference corpora that include multimodal spoken data in the form of audio recordings (e.g. *BNC2014*), topic-specific large text collections (e.g. *News on the Web*) and topic-specific learner corpora (e.g. the *Giessen–Long Beach Chaplin Corpus*). And, of course, there are other distinctions that could be added to the list, such as diachronic corpora and corpora of child language acquisition (e.g. Child Language Data Exchange System (*CHILDES*)). For a rough and incomplete overview meant to highlight this fuzzy nature of the categories, see [Table 1](#). As a consequence, rather than suggesting a fixed taxonomy-type list, we selected the distinctions to draw attention to those opportunities and challenges of corpus pragmatics where we see most potential for research developments at the moment.

In [Section 2.2](#), we start our discussion with a group of corpora that describes some of the oldest and still very common corpora, namely balanced reference corpora. Their counterpoint are topic-specific corpora, discussed in [Section 2.3](#). The next two sections deal with modality. In [Section 2.4](#), we describe corpora that are based on spoken language and in [Section 2.5](#), we discuss the multimodal representation of data in corpora. [Section 2.6](#) focuses on corpora that include pragmatic annotation. The following two sections deal with what language is represented in a corpus. Corpora on learner English are discussed in [Section 2.7](#), and [Section 2.8](#) introduces parallel multilingual corpora. The final two sections deal with size and data selection. Corpora that are based on unsystematic large text collections are discussed in [Section 2.9](#), and [Section 2.10](#) starts the discussion of purpose-built self-compiled corpora, which will be continued in [Sections 3](#) and [4](#). Each distinction is introduced briefly and a few examples of relevant corpora are mentioned. We also present some advantages and limitations of such corpora for pragmatic research and provide a few examples of pragmatic studies that have been carried out in the past.

The corpus landscape is changing quickly. For up-to-date information of existing corpora and their accessibility, electronic sources are far more suitable than printed publications. There is not a uniform point of access for all existing linguistic corpus resources, although various attempts at establishing repositories of corpus data and/or information about corpora have been made. Corpora can be found, for instance, in research infrastructures like the *Common Language Resources and Technology Infrastructure (CLARIN)*, online repositories like the *Oxford Text Archive (OTA)* and corpus managers like *Sketch Engine*.

**Table 1** Overview of a selection of corpora and the corpus categories they simultaneously belong to, ordered alphabetically according to the name of the corpus

Corpus categories→ Name of corpus↓	balanced reference corpus	topic- and domain- specific corpora	spoken corpora	multimodal corpora	pragmatically annotated corpora	learner corpora	unsystematic large text collections
BNC 1994	x		(x)	(audio)			
CHILDES		x	x	audio/video			
CLMET3.0		x					x
EEBO		x					x
Giessen–Long Beach Chaplin Corpus		x	x	audio		x	
ICE	x		(x)				
LINDSEI			x			x	
Longman Corpus of Spoken and Written English	x		(x)				
News on the Web		x					x
NMMC		x	x	audio/video			
SBC			x	audio	x		
SPICE-Ireland Corpus			x			x	
VOICE			x	(audio)			

All abbreviations of corpora mentioned in Table 1 can be found in the discussion in this section.

x: applies

(x): transcripts only

A valuable database of information about English language corpora is the *Corpus Resource Database (CoRD)*, which includes a corpus finder tool that filters corpora according to language period, word count, type of data and annotation, and availability.

## 2.2 Balanced Reference Corpora

Balanced reference corpora are corpora that include text samples from a variety of different contexts in order to provide an overall representation of a given language variety. One of the most frequently used reference corpora is the *BNC*. It includes different types of spoken and written language samples from British English. While the original corpus, the *BNC1994*, included data from the 1990s, the recently released *BNC2014* adds data from around 2014, thus providing opportunities for investigations into recent language change. Other examples are *COCA* and the *International Corpus of English (ICE)*. The latter consists of an entire collection of corpora that represent English varieties from around the world. Another group of reference corpora are the corpora used as a basis for compiling dictionaries and grammars, such as the *Longman Corpus of Spoken and Written English*. Important diachronic reference corpora of English are the *Helsinki Corpus of English Texts (HC)* and the *ARCHER* corpus, and, for American English, the *Corpus of Historical American English (COHA)*.

**Advantages:** Balanced reference corpora provide a general-purpose perspective on a language variety and are particularly well-suited for investigating general characteristics of a language variety. They are also very suitable for studying register variation, e.g. the variation of language phenomena across different text types and genres.

**Limitations:** Balanced reference corpora are usually compiled for facilitating the study of lexico-grammatical characteristics. They usually only support form-based queries and they often present decontextualised views of the data with limited access to context.

**Examples of previous studies:** Many form-to-function studies have been based on balanced reference corpora, e.g. the study of discourse markers (e.g. Aijmer 2002, 2008) and planners (Tottie 2011). Apologies have been studied in the *BNC* (Deutschmann 2003) and the *COHA* (Jucker 2018).

## 2.3 Topic- and Domain-Specific Corpora

Corpora can be compiled for any given topic or domain. Examples of recently released corpora include *SydTV*, based on transcripts of language used in television series; the *Brexit Corpus*, a 100-million-word collection of online

texts relating to the UK referendum on Brexit; and the *Corpus of Historical English Law Reports 1535–1999 (CHELAR)*. They are often compiled for a specific research project and, thus, their composition, annotation and suitability for pragmatic studies varies a great deal. While a single corpus only provides insight into one domain, different corpora can sometimes be combined to gain a broader perspective (see [Section 5](#)).

**Advantages:** Those topic- and domain-specific corpora that were compiled with pragmatic research questions in mind often offer annotation and contextual information that make them extremely suitable for pragmatic studies.

**Limitations:** Given that many domain-specific corpora were compiled for a specific research project, it can be challenging for other researchers to use these corpora for different projects. The composition, markup and metainformation may be geared towards some research questions rather than others. Consulting corpus documentations and learning about the corpus data is perhaps even more important than for other corpora. In some cases, it can be difficult to gain access to documentation or the information provided may be insufficient. However, some domain-specific corpora are also among the corpora with the most detailed amount of background information.

**Examples of previous studies:** There are many different kinds of examples that could be quoted here. [Lutzky and Kehoe's \(2017a, 2017b\)](#) studies of apologies in the *Birmingham Blog Corpus (BBC)* are an example of research based on contemporary data. Examples of studies on historical corpora are [Traugott's \(2015\)](#) study of interjections in the *Old Bailey Corpus (OBC)*, [Culpeper and Kyö's \(2010\)](#) study of pragmatic noise in the *Corpus of English Dialogues 1560–1760 (CED)*, and various studies on the development of scientific thought-styles that are based on the *Corpus of Early English Medical Writing (CEEM)* (e.g. [Gray, Biber & Hiltunen 2011; Taavitsainen 2001, 2002, 2009; Whitt 2016](#)).

## 2.4 Spoken Corpora

Early corpora that were based on spoken language data, such as the original *London-Lund Corpus (LLC)* often included only written transcriptions. In recent years, it has become more and more common to provide access to audio recordings, which are often aligned with the written corpus data, which makes them multimodal corpora (see [Section 2.5](#) and [Section 7](#)). Examples are the spoken section of the original *British National Corpus (BNC1994)*, for which recordings have recently been made accessible, as well as the recently released new corpus component *BNC2014* and the soon-to-be released *London–Lund Corpus 2 (LLC2)*. The *Santa Barbara Corpus of American English (SBC)* is another

important resource for the study of spoken language, consisting of sixty recordings and corresponding transcriptions. More specialised spoken corpora include the *Cambridge and Nottingham Business English Corpus* (*CANBEC*), *British Academic Spoken English Corpus* (*BASE*) and the *Vienna-Oxford International Corpus of English* (*VOICE*), which includes spoken language data from speakers of English as a lingua franca. In addition to differences in the availability of audio recordings, the corpora vary greatly with respect to the degree of detail and the amount of prosodic information that is included in the transcriptions.

**Advantages:** In addition to presenting data from one important mode of language realisation, spoken language corpora that provide access to audio recordings can give insight into how paraverbal features like loudness, stress, speech rhythm and other prosodic features affect pragmatic functions, an aspect that has not been explored in great detail so far (see also [Section 2.5](#)).

**Limitations:** The integration of written transcriptions and audio recordings poses challenges. The new *BNC* interface has made great progress in this respect, even making it possible to search for phonological phenomena. Still, search queries usually rely on written transcriptions and the transcription conventions vary a great deal across different corpora.

**Examples of previous studies:** [Adolphs & Carter \(2013\)](#) present a very detailed discussion of spoken corpus linguistics, which includes studies on pragmatic aspects such as response tokens.

## 2.5 Multimodal Corpora

Multimodal corpora are corpora which, in addition to text, also include data in different modalities. For instance, this can take the form of audio recordings of spoken language or visual information in the form of still images or videos. There are many different ways in which data in different modalities is presented. The most basic form can be found in corpora that include separate files with the different modalities, such as the *SBC*. The *SBC* consists of a collection of discourse analytic transcriptions of spoken language recordings, and the recordings are provided as separate audio files. At the opposite end of the scale we can find corpus infrastructures that include multilayered annotation of data aligned across different modalities. An example of this latter type is the *Nottingham Multimodal Corpus* (*NMMC*), which includes video recordings, transcriptions of the spoken language and the annotation of non-verbal communication like gestures.

**Advantages:** Multimodal spoken corpora make it possible for researchers to study verbal, paraverbal and non-verbal features of language in combination.

In written language, the combination of text with images, videos, layout and typographic elements can be studied. Thus, multimodal corpora open up new avenues of pragmatic research.

**Limitations:** By far the biggest limitation of multimodal corpora is that their compilation is very time-consuming and, as a consequence, they tend to be rather small. Corpora that include layers of annotation for non-textual information, such as gestures or eye-gaze, face the same challenges as transcriptions of spoken language, namely that the transcription of such information involves a considerable deal of qualitative interpretation.

**Examples of previous studies:** We discuss different examples of studies based on multimodal corpora in Section 7.

## 2.6 Pragmatically Annotated Corpora

Some corpora include annotation that is relevant for pragmatic analysis (for an overview, see O’Keeffe 2018: 599–605). In some cases, corpus annotations can inform pragmatic analysis, whereas in others, it is pragmatic information that is annotated. For instance, the *Sociopragmatic Corpus (SPC)* is a subsection of the *CED*, in which speaker information such as gender, social status, role and age group is annotated on the level of each utterance. This makes it possible to study, for instance, how the role relationship between interlocutors affects the realisation of pragmatic functions. An example of annotation of pragmatic information can be found in the *SPICE-Ireland Corpus*, a subsection of the spoken component of the *ICE-Ireland* corpus that is annotated for pragmatic and prosodic information. The annotation includes information about the speech-act function of utterances, discourse markers and quotatives.

**Advantages:** Depending on the kind of annotation, annotated corpora make it possible for researchers to search for pragmatic functions or to restrict their queries to factors that are relevant for pragmatic analysis.

**Limitations:** Since annotation is time-consuming, these corpora tend to be small. Attempts at automatising the process have been made (see Weisser 2014 on the automatic annotation of speech acts), but it is doubtful whether the process of manual classification will ever be fully replaced and, thus, the time needed to add annotations to a corpus will likely continue to pose limits to the size of annotated corpora. Moreover, studies are restricted to the kinds of information that have been annotated in the corpus, and given that the annotation of pragmatic information already involves a great deal of interpretation, the corpus compiler’s assessment of the data influences the results.

**Examples of previous studies:** Kirk's (2015) study of the pragmatic markers *kind of* and *sort of* is based on the *SPICE-Ireland Corpus*. The *Sociopragmatic Corpus* has been used for a range of studies on pragmatic features in Early Modern English, including Archer's (2005) studies of questions and answers, Culpeper and Archer's (2008) study of requests and Lutzky's (2012) study of discourse markers.

## 2.7 Learner Corpora

Learner corpora include language produced by non-native speakers of a language. Such corpora usually provide information on the level of proficiency and/or the amount of instruction of the speakers or writers that contribute data. Examples of learner corpora are the *International Corpus of Learner English (ICLE)*, based on essays from learners from a large number of different language backgrounds, and the *Louvain International Database of Spoken English Interlanguage (LINDSEI)*, the spoken counterpart to *ICLE*.

**Advantages:** Learner corpora make it possible to study pragmatic aspects of L2 language learning. They can also provide new perspectives for language teaching.

**Limitations:** The data in learner corpora is often derived from task-based language production. While written learner corpora may include essays that were produced in the ordinary course of the teaching process, spoken corpora often include task-based interaction that took place for the purpose of corpus composition, and that sometimes also includes language produced by interviewers involved in corpus composition. This distinguishes learner corpora from most other kinds of corpora, which tend to include language that was produced without elicitation for research purposes.

**Examples of previous studies:** Examples of corpus studies of pragmatic features in learner corpora include the study of hesitation markers (Gilquin 2008) and discourse markers (Buyssse 2012, 2017, 2020) based on the *LINDSEI* corpus. Another example is Huschová's (2021) study of the pragmatic functions of *can* and *could* in speech acts in the *Corpus of Czech Students' Spoken English*.

## 2.8 Parallel Multilingual Corpora

The term 'parallel corpus' refers to a corpus of specific design, which includes related subcorpora aligned in respect to one another according to certain criteria (these might be structural, e.g. sentence or timestamp alignment, or semantic, e.g. translation unit alignment). A multilingual parallel corpus contains source texts, or transcripts, in one language, and their translation(s) into other language(s).

The parallel aligned design makes it possible to locate an excerpt in the source text and its corresponding translation using a single query. Various parallel corpora are available to date (see Examples of previous studies). For researchers interested in compiling their own parallel multilingual corpus, SketchEngine offers a corpus infrastructure that supports this.

**Advantages:** Parallel corpora enabled a qualitatively new step in the studies of translation and cross-linguistic analysis, finally making it possible to test empirically the assumptions about translation universals, improve machine translation and provide translators with a more reliable, up-to-date and usage-based alternative to a conventional dictionary.

**Limitations:** Parallel corpora are notoriously difficult to compile since they require an additional step of aligning the subcorpora. This has prompted calls to reuse existing parallel corpora rather than invest in the creation of new purpose-built ones (Doval & Sánchez Nieto 2019). In addition, professional translators in large organisations often rely on shared translation memories such as DGT-TM, i.e. organisation-specific databases that store sentences, paragraphs or segments of text that have been translated before. This makes the resulting translation corpora uninteresting for research in contrastive linguistics (Cartoni et al. 2013).

**Examples of previous studies:** The key resources of this type, such as the *EUROPARL* (the proceedings from the European Union in twenty-one languages, see Koehn 2005) or *OPUS* (subtitles and localisation resources covering 200 language variants, see Tiedemann 2012), generated hundreds of studies. They range from classic issues of translation studies, e.g. testing the explicitation hypothesis (Zufferey & Cartoni 2014), to research questions in contrastive linguistics, e.g. studying how impersonalisation is expressed in English versus German (Gast 2015).

## 2.9 Unsystematic Large Text Collections

There are a number of interfaces that make it possible for researchers to search collections of often unannotated electronic documents that were not systematically selected to build a linguistic corpus. A prominent example is the *Google Ngram Viewer*, which can be used to compare the frequencies of word sequences in printed (and electronically available) sources since 1500. Similarly, *EEBO* provides a search interface for almost 150,000 titles of historical printed books from 1475 to 1640. With *EEBO*, researchers have the option of searching for some amount of metainformation on the books, as well as of viewing images of page scans of the book. However, there is only limited access

to the full text of *EEBO* and more advanced corpus operations – e.g. search for collocations and keywords – cannot be carried out. Similarly to *EEBO*, the *Corpus of Late Modern English Texts (CLMET 3.0)* provides access to the full books published by British authors between 1710 and 1920.

**Advantages:** The main advantage of such search interfaces is that they often provide access to a massive amount of data. This is especially valuable for studying less frequent linguistic phenomena, for which large corpora are needed to find a sufficient number of attestations. In addition, large text collections can provide some insight into overall diachronic developments, especially when different expressions are compared to each other. Finally, large amounts of data are ideal for the training of machine learning language models, which can be used for the computational analysis of linguistic features, including pragmatic ones.

**Limitations:** Unsystematic text collections do not have the same degree of reliability as smaller, manually compiled corpora. Documents can be misclassified, leading to wrong results, and automatic text recognition, which is often used in preparing the data, can misrepresent the content of documents. These problems can be reduced by editing the data, but due to the data size, this process is often time-consuming. In addition, contextualisation is often a problem. For instance, the *Google Ngram Viewer* only provides frequencies of word sequences, but there is no option of viewing instances of the sequence in context. These limitations mean that such data sources can usually only provide one piece of information, which has to be complemented with additional evidence through triangulation of methods.

**Examples of previous studies:** The *Google Ngram Viewer* has been used in various studies of metacommunicative expressions, such as in [Jucker and Kopaczyk's \(2017\)](#) account of historical (im)politeness. Likewise, [Jucker \(2020\)](#) presents results from *CLMET*, in addition to the *Google Ngram Viewer*, in tracing politeness expressions across time.

## 2.10 Purpose-Built Self-Compiled Corpora

Many corpus-based studies of pragmatics are based on corpora that are purpose-built for the study in question. Depending on the project for which they are created, such self-compiled corpora can vary greatly in size, composition and metainformation that is included. They can take the form of small, carefully selected and richly annotated collections of data as well as vast compilations of documents that are electronically available. Likewise, they may or may not be made accessible to other researchers (see also Limitations).

Self-compiled corpora can be processed with generic corpus software, which includes freeware such as *AntConc*, *TXM* and the tools offered by the *IMS Open Corpus Workbench (CWB)*, as well as paid solutions like *WordSmith Tools* or *Sketch Engine*. For smaller projects and qualitatively oriented research questions, tools for qualitative data analysis can be used, such as *MAXQDA*, *NVivo* and *ATLAS.ti*. For researchers comfortable with using programming languages, additional tools are available, such as *Python's Natural Language Toolkit (NLTK)* and *spaCy*, or *R's quanteda* and *koRpus*. For many purposes, a good text editor and pattern-matching with regular expressions can be sufficient to help researchers process text-based data. For more advanced analysis, XML-based corpora can be annotated and searched with XPath or CSS Queries, for instance using the *Oxygen XML Editor*. *CWB* corpora can be hosted on a server and made accessible via a graphic user interface (*CQPweb*).

**Advantages:** Purpose-built corpora present exactly the kind of data that is needed to answer a given research question. In addition, researchers who compile their own corpora tend to be extremely familiar with the data that is included, which is of great advantage for interpretation.

**Limitations:** The biggest drawback of self-compiled corpora is probably the amount of work needed for compiling them. This is one of the arguments in favour of making corpora accessible to other researchers, so that they can benefit from the work as well. Another argument is transparency. Only by providing data access to other researchers can results be verified independently. The FAIR Principles ([www.go-fair.org/fair-principles/](http://www.go-fair.org/fair-principles/)) discuss the relevant considerations behind this. Moreover, researchers need to deal not only with technical issues, such as suitable data repositories, standards of file formats and annotation, but also with a large range of ethical and legal questions concerning data collection and sharing (see [Section 3.3](#)). The latter concerns are, of course, also an issue for the corpus compilers of other corpus types but no longer for the users of the finished corpora, who rely on ethics and copyright having been taken care of by the compilers.

**Examples of previous studies:** We discuss different examples of studies using purpose-built self-compiled corpora in [Sections 3](#) and [4](#).

## 2.11 Conclusion

The outline of corpus categories presented in this section makes it clear that corpora come in all kinds of sizes, forms and structures, and with a wide variety of different kinds of contents. As a consequence, corpus pragmatics is far from a homogenous field of research. Methods that are well-suited for searching text

in balanced sampler corpora cannot be applied directly to the analysis of gestures in multimodal corpora, and the skills needed by researchers for compiling and annotating their own handcrafted datasets will differ vastly from the skills needed to work with interfaces to unsystematic large text collections. Thus, any discussion of corpus pragmatic methods needs to first establish what kind of corpus data is involved in a given study. In the remainder of this Element, we take a closer look at some types of corpora and the kinds of challenges and opportunities they present for research.

### 3 Corpus-Driven Approaches with Self-Compiled Corpora

#### 3.1 Introduction

The distinction between corpus-based and corpus-driven approaches to linguistics was introduced by [Tognini-Bonelli \(2001\)](#), who discussed the underlying assumptions behind corpus work with language data. She posited that while corpus-based studies use a systematically collected dataset – a corpus – in order to test or explore a pre-existing hypothesis, corpus-driven studies claim that the corpus itself is the sole source of hypothesis about language. A corpus-driven approach to pragmatics, therefore, treats a corpus as more than a repository of examples or a reference for a bottom-up development of evolving taxonomies (as with the methods described in [Section 4](#) of this Element).

While corpora are often created with the aim to allow analyses relatively free from preconceived notions about language, it is important to note that corpus building is itself an analytical step and that no corpus is theory-free. This is especially true for corpus-driven pragmatics, since pragmatic meaning is necessarily more interpretative than, say, lexicogrammar. In order to remove researchers' bias as far as possible, corpus-driven approaches rely on automatic annotation of pragmatic phenomena. As such, corpus-driven research requires relatively large corpora and sophisticated automatic annotation. Pragmatic annotation layers may include classic pragmatic units (speech acts, deixis) and units on the interface between pragmatics and semantics (such as contextualisation cues or positive versus negative polarity). Some scholars draw on grammatical annotation for corpus pragmatics as well, arguing that the creation of communicative meaning involves all levels of language (for instance, [Rühlemann & Clancy 2018](#), who study indicative and subjunctive verb forms as an expression of deixis).

The work on (semi)automatic speech-act annotation of English language corpora includes the *Speech Act Annotated Corpus* project (*SPAAC*, University of Lancaster, [Leech & Weisser 2003](#)) and the implementations of the *Dialogue Annotation & Research Tool* (*DART*, [Weisser 2010, 2015](#)). Of course, some speech

acts lend themselves better to automatic annotation than others. For example, congratulations are strongly associated with predictable surface forms such as ‘congratulations’, ‘happy birthday’, ‘merry Christmas’ – the so-called IFIDs. The bulk of corpus pragmatic research within the corpus-driven approach has been done on speech acts with a stable list of IFIDs, for example, apologies (Lutzky & Kehoe 2017a; Jucker 2018). Some ambitious attempts in natural language processing aimed to include more contextual factors when training an algorithm to recognise speech acts that do not include IFIDs – for example, flirting (see Ranganath et al. 2009).

In this section of the Element, we zero in on a special case of corpus-driven pragmatics: when researchers rely on fully automatic extraction of linguistic features but steer their interest through the process of corpus compilation. Much of automatic corpus analysis relies on the comparison of subcorpora (for example, keyword analysis, contrasting lists of n-grams or ranking collocates of an item). This means that through corpus design, researchers can manifest their interest in specific variables, while keeping the study of those variables entirely corpus-driven. For example, compiling a corpus of book reviews with different reader ratings enables the researcher to analyse how polarity and evaluation differ depending on whether a book is liked (Rebora et al. 2021: ii240–1).

In this section, we will discuss the process of creating such a corpus, and the benefits and challenges of this type of corpus pragmatic research, based on the example of the r/changemyview study of persuasion. The challenges and choices of the corpus creation process that are discussed in this section are not unique to corpus-driven pragmatic research and can be usefully applied in other research contexts, including those described in Section 4. The present corpus was created by downloading posts and replies from the popular online forum website Reddit ([www.reddit.com/](http://www.reddit.com/)) (See Dayter & Messerli 2022). Reddit is a platform subdivided into thousands of thematic forum communities, many of which lend themselves well to the study of specific pragmatic phenomena. Below, we discuss a purpose-built corpus of the forum threads from the r/changemyview (*CMV*) subreddit ([www.reddit.com/r/changemyview/](http://www.reddit.com/r/changemyview/)), a community devoted to persuasive practices.

## 3.2 Case study: Persuasion in a Corpus of Reddit Posts

### 3.2.1 Designing a Corpus for the Study of Persuasion

Persuasive discourse has been defined in linguistic research as all linguistic behaviour that attempts to either change the thinking or behaviour of an audience, or to strengthen its beliefs, should the audience already agree (Virtanen & Halmari 2005).

Since the judgement of whether a stretch of discourse is persuasive or not (i.e. has changed or strengthened the readers' views) depends on the speaker's or writer's intent, the decision is ultimately interpretative and would have to be realised through the process of manual annotation.

However, through the careful choice of data and corpus design, researchers can enable a corpus-driven pragmatic study. The first step in this process is to find a dataset that contains persuasive discourse, recognised as such through both interaction-internal and interaction-external orientation. A good example of data of this kind can be found on the subreddit r/changemyview. Not only does the subreddit self-identify as a community dedicated to changing users' views, comments posted on it also include several language-external indicators of popularity and persuasiveness. One of these indicators, called karma, is Reddit's equivalent to upvotes and likes on other platforms and social networks and points to the uptake of individual posts by other members of the community. Collaboratively, the community thus signals their approval by adding to the post's and poster's karma. In addition, and more importantly for the persuasion-oriented researcher, threads of comments on *CMV* award a great deal of interactional power to original posters (henceforward OP). OPs first post a statement with an invitation for others to change their, the OP's, view. When comments are posted, OPs then decide which of them succeeded in persuading them and reward such convincing posts with a delta ( $\Delta$ ), which signals to others that the OP found the arguments in the awarded post persuasive. Consequently, the researcher who collects a corpus of *CMV* data and includes delta awards as metadata, can structure their data so that successfully persuasive comments (delta-awarded comments, DACs) and unsuccessful attempts at persuasion (non-DACs) can be contrasted. Without the limitations of automatic annotation of pragmatic features, or the etic judgement of one or several trained coders, it is thus possible to create a large corpus of attempts at persuasion and subcorpora of more and less persuasive comments (Dayter & Messerli 2022).

### 3.2.2 Building a Corpus: Ethics and Copyright

The first concern of any empirical project is the availability and researchability of data, which means that researchers must inevitably ask themselves first whether the data they want to study can be acquired in a manner that is both ethically sound and in agreement with existing copyright laws. While smaller qualitative studies primarily establish this by getting the explicit consent of the rights holders to the data they use (typically the authors themselves), research based on larger corpora cannot follow the same pattern, since it is not feasible to ask, say, hundreds of thousands of *CMV* users for permission to use their data.

This has been recognised in recent times also by new legislation, e.g. in Germany and Switzerland, where some of us work, which specifically allows big data research without consent for scientific purposes. Following these laws is the first step in ensuring sound research, but it does not exempt researchers from carefully assessing the ethical ramifications of their research for the people whose work they are scrutinising. Just like in the case of studies based on consent, researchers need to also ask themselves whether their studies will be beneficial or could potentially be harmful for authors represented in the corpus as well as for society at large. A useful guide for interpreting the existing ethics guidelines with respect to corpus linguistic approaches can be found, for example, in [Koene and Adolphs \(2015\)](#), focusing on internet data, or the AoIR Internet Research Ethics 3.0 ([franzke et al. 2020](#)), addressing specifically the problem of informed consent in big data analyses.

In the example we choose here, the researchers established that the population chosen for research is not vulnerable and the textual data does not concern issues that might cause potential harm if disclosed in publication. Moreover, the subreddit r/changemyview encourages academic research and has a page dedicated to academic papers using *CMV* data. Therefore, there is no potential harm to social media users, or potential ethical problems associated with the project.

In addition to ethics concerning those whose work is included in the corpus, there are further ethical concerns and responsibilities towards the scientific community. In this sense, ethical research always attempts to be optimally transparent and to facilitate reproducibility of its result. It is worth pointing out, however, that ethical concerns for the research community and for the researched population are often at odds. Researchers find themselves confronted with the uncomfortable choice of increasing the potential for harmful outcomes for the communities they research or reducing the transparency of research, with the latter typically considered the lesser evil. Thus, the consensus in research ethics is that social media posts should not be redistributed as datasets, since this denies agency to individuals who have subsequently deleted their posts and may be entirely unaware that their data are circulating in third-party datasets shared among researchers (see [Proferes et al. 2021](#)). This stance is fixed both in the Twitter's Terms of Service and, more generally, in European GDPR's stipulation on the right to be forgotten. The compromise to satisfy the demand on reproducibility is to create 'dehydrated datasets', i.e. lists of unique post IDs instead of files with full textual data and metadata.

### *3.2.3 Building a Corpus: Technical Challenges*

A corpus in the sense we are using it here is typically an operational representation of the linguistic practices of a community. In our example, the researchers

considered the subreddit *CMV* as a speech community whose interactions, while being perhaps similar to others on Reddit, in social media and elsewhere, can reasonably be modelled in a research corpus for internal as well as contrastive studies. Since the corpus is thus in a particular relationship to that which it models, i.e. to the utterances that were and are accessible on r/changemyview, researchers are faced with decisions about how to represent the population in the corpus. Choices include such aspects as:

- The amount of data – how many posts should be included in the corpus?
- Which texts, if any, should be excluded from the corpus – e.g. duplicates
- The representation of each text, including typography, aesthetic presentation, and linguistic, paralinguistic, and non-linguistic context (see [Section 7](#) for considerations concerning multimodality)
- The inclusion of internal and external metadata – e.g. delta, the time of posting or information about the individual posters
- The hierarchy of and relationship between texts in the corpus
- Additional annotations that should be added by the researchers to facilitate their research

In the case of the *CMV* corpus, the researchers chose to collect all posts that had ever been posted to r/changemyview, which means that it would include all posts between May 2013 and the time of data collection, May 2020. When it comes to excluding texts, corpus-building choices reflected what is and is not considered to be a part of *CMV* communication: in some cases, posts marked as deleted can still be accessed through the pushshift.io Reddit API that was used to collect the data, as are those that are marked as being in breach of the subreddit's rules. Both these types of posts were excluded from the final corpus. In addition, the cut-off date points to a problem concerning the fixity of the text ([Jucker 2004](#)). While the raw data of social media in particular may grow indefinitely (submissions to r/changemyview written before 2020 may still receive comments years later), the static representation in the corpus is often fixed at the time of collection, thus forcing dynamically evolving discourse into a rigid snapshot. Corpora such as Davies' *NOW (News on the Web)* corpus may instead be updated at regular intervals and thus be conceptually dynamic. Since published research is largely static at the time of rewriting, a more dynamic approach creates issues in terms of reproducibility, and in the inevitable choice between the two strategies, the researchers building the *CMV* corpus thus chose stability of results over representation of dynamic data.

In terms of the representation of the texts, the *CMV* corpus only included verbatim the roughly six million texts themselves, but not their visual organisation into threads, no audiovisual materials users may have posted alongside

their texts (of which there are very few if any), and no typographic and other aspects of the graphic appearance on Reddit. The corpus includes metadata that is also visible on r/changemyview, like the time of posting, the delta status of posts – were they awarded a delta or not? – and a unique user for each comment. However, for ethical reasons all user names were anonymised. The relationship between texts was also represented as textual metadata, with hierarchies flattened into annotations as to what threads comments belong to and what posts or comments they respond to. Finally, regarding additional annotations, the corpus was lemmatised and tagged for parts of speech.

While the *CMV* corpus exists in an initial form, encoded in the *CWB* (see Evert & Hardie 2011) and accessible locally through command line tools as well as using the *R* package *polmineR* (Blaette 2020), it is important to point out that individual studies will typically use corpora in particular ways, which often amount to actually or virtually creating new corpora or subcorpora and thus new representations of aspects of communication in the original population. In the *CMV* project, an initial focus was a contrastive study of linguistic differences between more and less successfully persuasive posts (see Section 3.2.4). Accordingly, the corpus was subsampled into two subcorpora based on the delta-marker: a 14.5 million word delta subcorpus (54,000 texts) was compared to a 133.7 million word non-delta corpus (500,000 texts). In this case, the delta subcorpus contains all comments available in the corpus and thus on r/changemyview (in May 2020), whereas the non-delta segment contains a sizeable sample of the population. This sampling of the non-delta subcorpus included the conscious choice to increase the similarities to the delta subcorpus, which meant that very short and very long texts were included until an almost identical average word length was achieved (266–7 words per comment).

### 3.2.4 Analysis of Formality and Persuasion

Based on the corpus design we illustrated above, the researchers in the *CMV* project conducted two studies targeting register differences in the *CMV* corpus. The detailed results of the first study are published in Dayter and Messerli (2022), with the second study forthcoming. Rather than reiterating these results here in full, we will offer a brief summary and use Section 3.3 primarily to highlight the insights the studies offer for corpus-driven research more broadly.

Using a comparative subcorpus design, we have investigated the linguistic differences between delta and non-delta comments along two dimensions: formality-informality and overt expression of persuasion. To operationalise the linguistic exploration of the delta/non-delta distinction inherent in the population and the corpus, procedures established and documented in the

existing literature were used. The first study identified twelve quantifiable linguistic markers (for example, frequency of WH questions and of nominalisations) that are associated either with high-formality or low-formality language. The second study drew on [Biber's \(1988\)](#) variationist analysis, which has proposed and empirically validated seven linguistic features associated with the overt persuasion dimension of variation (such as frequency of infinitive, prediction modals or suasive verbs).

After a statistical analysis (the chosen method here was the Mann–Whitney *U* test, a test that can be used to compare data that is not normally distributed), we arrived at the conclusion that there are no significant differences of high or medium effect between the delta and non-delta posts. This was true both for the formality and for the overt persuasiveness dimensions. The studies fell squarely into the category of negative results – a situation when the anticipated hypothesis is not borne out by the corpus data.

Taken in isolation, the results of this corpus-driven study did not contribute in a significant way to our understanding of persuasive language on *CMV*. However, it is exactly in this context that the benefits of combining corpus-driven methods with a closer inspection of the language data become apparent. To triangulate the quantitative findings, we designed our study to also include manual analysis of the concordances of two category-bound actions: *to persuade* and *to change someone's view*, as well as a study of the highest-frequency n-grams. This closer look at community discourses revealed an explicit orientation towards how, in the opinion of community members, a hypothetical 'good user' or 'good persuader' should write. The users frequently referred to the binary distinction between a personal opinion that cannot be changed by facts and evidence (an undesirable stance) versus a malleable view that can be disproven in an objective manner (desirable stance). The two examples below illustrate this community norm:

- (3.1) We can't really change your view of a personal opinion. If you don't enjoy ET then you don't enjoy ET.
- (3.2) First, you can change your view multiple times, and you claimed secular buddhism is the most common, and I disproved that with numbers.

As [Example 3.2](#) shows, 'change your view' is often understood as 'provide evidence that your view is incorrect'. The members of the *CMV* community recognise formal register as the appropriate linguistic resource to draw on in evidence-oriented discourse. Taken together, the corpus and the qualitative analytical findings confirmed that *r/changemyview* has an established linguistic community norm: all users write in a formal, overtly persuasive style, perhaps reflecting the community's affinity for academic-like argumentation.

### 3.3 Conclusion

Corpus-driven approaches to pragmatics aim to be as unbiased by pre-existing assumptions about the dataset as possible. This stance is very useful when working with pragmatic categories that can boast robust form-to-function correlation, for example deixis or discourse markers. However, adding an interpretative component in an analysis of higher-level ‘strategies’ or ‘moves’ inevitably requires a departure from an approach to data completely uninformed by any pre-existing theory. In this section, we demonstrated how a compromise may be achieved through careful preliminary corpus design, which is then used as the sole source of hypothesis about language in line with the corpus-driven pragmatics setup. Despite the overall success of the approach, some interesting observations resulting from the study could only be put in sufficient context with help of a qualitative close-up look. The [next section](#) of the Element, [Section 4](#), will continue down this avenue of research to describe the benefits and limitations of corpus-based and corpus-assisted approaches to pragmatics.

## 4 Corpus-Assisted Approaches with Self-Compiled Corpora

### 4.1 Introduction: Corpora in Discourse Analysis

Corpus pragmatics is an important field of research that enables linguists to analyse public discourses about societal issues, such as gender politics and gender relations ([Lutzky & Lawson 2019](#)), voting behaviour ([Ifukor 2010](#)), right-wing activism and misogyny ([Krendel et al. 2022](#)), terrorism ([Harrison et al. 2008](#)) or the anti-vaccination movement ([Quo VaDis Project 2022](#)). However, traditional corpus-driven methods (such as the ones described in [Section 3](#) of this Element) may lack the human analyst’s judgement, which is highly important for understanding discourse. The gap between an automatic analysis of corpora of moderate to large scale and a close reading of each example by a human can be bridged by corpus-assisted approaches to pragmatics and discourse analysis.

More qualitatively oriented researchers make use of a number of corpus-assisted techniques that harness the power of comparatively large datasets in order to find patterns, quantify them descriptively or even simply check the presence of posited forms in the data. Perhaps one of the most prominent approaches one should mention is corpus-based discourse analysis, or CBDA (see, e.g., [Baker 2006](#)). A common criticism of close-reading approaches to discourse analysis is that they may be too deeply rooted in the researcher’s pre-existing belief. CBDA makes a step towards the benchmark of empirical, unbiased investigation by relying more on quantified and reproducible findings.

Heinrich and Schäfer (2018: 135) emphasise the auxiliary role of the corpus, which should ‘enable hermeneutic researchers to analyze and qualitatively interpret huge amounts of textual data without excessive cherry-picking’. Corpus-based discourse analysis typically involves obtaining concordances of search terms using corpus software, and manually analysing and interpreting them to uncover *discourse*: ‘a set of meanings, metaphors, representations . . . that in some way together produce a particular version of events’ (Burr 1995: 48). This has been done for various issues and employing different datasets: some examples among many are Koteyko et al. (2008) for infectious diseases in UK media and government discourses, Dayter (2016) for self-representation of ballet apprentices online, Kim (2014) for the representation of North Korea in US media, Messerli and Locher (2021) for humour support in timed comments and Dayter and Rüdiger (2022) for manosphere discourses.

Another step towards the researcher’s pre-existing taxonomies informed by literature, intuition and pilot studies is made in the family of analytical techniques known as corpus-assisted discourse studies (CADS). CADS research often relies on the critical discourse analysis framework and addresses politically laden issues, such as construction of race (Krishnamurthy 1996), discourse about the Iraq war in the press (Marchi & Taylor 2009) or representation of European citizenship (Bayley & Williams 2012). CADS differs from CBDA in respect to the degree of presuppositions and taxonomies that the researcher brings to the analysis. Love and Baker (2015), for instance, studied British Parliamentary arguments against LGBT equality and how they evolved over time. On the one hand, the method they used is very similar to classic CBDA. They started with a keyword analysis and a search for the collocates of the relevant terms to identify the salient lexical items in the debate, and then manually examined the concordances for these items. On the other hand, the choice of the search queries was informed by the purpose of the study: analysing the gradual shift from more to less explicit homophobic argumentation in the debates about same-sex marriage. Love and Baker (2015: 84) describe how their pre-existing agenda informed the study: ‘knowing that the speakers . . . voted against gay marriage, and that public attitudes towards homosexuality had altered enables a fuller interpretation and explanation of their language’.

Corpus-assisted approaches to pragmatics and discourse analysis present a multidimensional space, ranging from methods that combine traditional lexical corpus searches with qualitative analysis along pragmatic dimensions. While some scholars identify strongly with labels such as CADS or CBDA, the borders between the approaches are often fuzzy. In fact, different steps are often combined in a sequential or simultaneous manner in order to design the best methodology to answer a particular research question. In what follows, we will

illustrate some of the possible combinations of methodological steps that allow the combination of qualitative and quantitative approaches in custom-made corpora. The advantage of this combination lies in arriving at patterns that are particular to the corpus in question by combining the results of different layers of analysis.

## 4.2 Case Studies: Scaffolded, Quantified Analysis of Online Health Practices

Scholars have many options when attempting to develop the best possible research design for their particular research interest. The examples we draw on here stem from research output that comes from a project entitled *Language and Health Online: Typing Yourself Healthy* (SNSF-funded<sup>1</sup>), which worked with custom-made corpora. The combined methods are:

- (1) a content/thematic analysis, which establishes the aboutness and context of the texts,
- (2) a discursive moves analysis, which establishes the compositional structure of the texts,
- (3) a selective analysis of linguistic form and function of particular linguistic expressions in context,
- (4) drawing on any type of further research steps that help us answer our research questions in more depth.

The first three steps (and the fourth to different degrees) are based on qualitative analysis that follows after codebook development and training in order to arrive at reproducibility and reliability of pragmatic patterns. With the help of corpus annotation software and corpus linguistics search methods, different layers of analysis can be brought together, as explained in what follows. The aim is to identify pragmatic patterns of communication that are valid for the corpus in question. This combination of methods is meant to illustrate only one possible approach and is not meant to claim exclusivity or novelty per se (since each of these steps exists independently from each other). However, this research design allowed us to arrive at meaningful insights about the practices in question which, crucially, also facilitated comparability and reproducibility.

To explain why the suggested steps and scaffolded approach was chosen for the *Language and Health Online* project, we first introduce its research aims and context. The general perspective we took was that of interpersonal

---

<sup>1</sup> We wish to thank the Swiss National Science Foundation for funding the project *Language and Health Online* (100016\_143286/1).

pragmatics (Locher & Graham 2010), i.e. the study of how it matters who speaks to whom; what type of and how relationships are created, maintained and challenged through language use; and how these patterns can be linked to intersecting ideologies (such as gender, politeness, age). What is particularly at stake in communication in health contexts is the constructed and perceived expertise (be this of a biomedical and/or experiential kind) and the potential creation of trust (Locher & Schnurr 2017). In 2006, Locher published work on an online health-advice column, working with a custom corpus of 2,286 question–answer letters in an online archive of the site (990,000 words) (Locher 2006). From 2012–17, a project continued this research avenue by working on complementary datasets. Rudolf von Rohr (2015, 2017, 2018) explored the role of persuasion in a corpus on smoking cessation, which comprised thirty professional anti-smoking websites as well as peer-to-peer online fora for cessation support. Thurnherr (2017, 2022) worked on a smaller dataset of email counselling in short-term therapy to explore the creation of the therapeutic alliance. The data consisted of the full short therapy cycle of five clients with one counsellor. What these projects have in common is a joint research interest in exploring identity construction and the creation of expertise within interpersonal pragmatics (Rudolf von Rohr et al. 2019), and the application of a mixed methodology that combines insights from corpus linguistics with qualitative analysis inspired by the study of relational work. From a corpus linguistic perspective, all three corpora are small and custom-made, but they are large enough to warrant a quantified approach in order to arrive at the representative patterns for the individual corpora. The motivation to engage in quantification of qualitative analytic steps stems from the desire to achieve a more robust result of the general patterns within the corpus. The methodological orientation of the *Language and Health Online* project was described in Locher and Thurnherr (2017) after the completion of the project. In what follows, we reiterate and complement the observations made there in light of the purpose of this Element.

As our research questions on relational work in e-health needed a qualitative interpretation of linguistic surface structures in context, we proposed a layered analysis of data in order to answer the following questions (Locher & Thurnherr, 2017: 17):

- (1) What characteristic activities are employed in the different e-health practices (e.g. conveying information, giving advice or reflecting on interactants' interpretations of events or relationships, inviting introspection)?
- (2) What linguistic strategies are employed to achieve these activities?

- (3) What is the relation between the patterns of linguistic strategies and the creation of interpersonal effects (e.g. solidarity, empathy, power, the therapeutic alliance, identity construction)?

In order to embed our analysis of the function of linguistic strategies (and not just their forms) in their context, we approached the corpora with the four steps mentioned at the beginning of this section: (1) a content/thematic analysis, (2) a discursive moves analysis, (3) a selective analysis of linguistic form and function of particular linguistic expressions in context and (4) drawing on a selection of further methods depending on the research question.

To illustrate this, first consider the analysis of *Lucy Answers*, a US health-advice column run by an educational institution. The data in this analysis consists of online advice entries in the form of a question letter by an advice-seeker and a response letter by an (invented) agony aunt. Table 2, reproduced from Locher (2006: 117), shows how the different layers of the analysis of the response letters are scaffolded. All of the levels were annotated with *Oxygen XML Editor*. The labels are mutually exclusive. In case of co-occurrence, combined tags were created (such as bonding-humour).

Step 1 was concerned with the overall topic of the entry, or the aboutness of the texts (on content/thematic analysis, see Guest et al. 2012; Saldaña 2013). In this case, the topic classification of the original data source was adopted from the archive that the source provides. This information later allowed for grouping of topics into biomedically oriented questions and more interpersonal-oriented topics.

Step 2 consisted of an analysis of the entire texts into their compositional units (often roughly one or more paragraphs) that contained so-called discursive moves, i.e. the ‘kind of contribution that the entry made to the ongoing interchange’ (Miller & Gergen 1998: 192). Rather than coding on the speech-act and sentence level, entire passages were coded in their overall contribution to the text composition (e.g. advice-giving, assessments, etc.). As advice-giving was the centre of analysis, it was then possible to see how the sequence of discursive moves prepares for advice-giving and to assess how this discursive move is embedded within the overall rhetoric of the text.

In Step 3, a selection of linguistic forms was systematically looked for and coded for their function. Since exploring advice-giving was the main aim of this study, an analysis of how the advisory sentences were syntactically realised (imperative, declarative, interrogative) was conducted (this was done within the advisory discursive move with an XML tag). In addition, a selection of relational work strategies were systematically tagged, such as hedging or boosting. The codes were partly adopted from existing literature but also developed

**Table 2** Categories for the analysis of problem and response letters of *Lucy Answers* (abbreviated and merged from [Locher 2006: 117, 209](#))

Step 1	Step 2	Step 3	
Topic category	Content structure level	Relational work level	
drugs	<i>Problem letter:</i> address		
emotional health	unit (one or more)	apology comment on previous record compliment explanation metacomment background problem statement question request advice thanks	appealing bonding boosting criticising hedging using humour in combination with the other relational work levels
fitness/ nutrition	pseudonym		appeal humour neutral none
general health			
relationships			
sexual health			
sexuality			
	<i>Response letter:</i> address		
	unit (one or more)	advice assessment disclaimer explanation farewell general information metacomment open category own experience referral	bonding boosting criticising empathising hedging praising using humour in combination with the other relational work levels
	signature		

bottom-up. The interest was to find out which discursive moves contained which relational work strategies. One of the findings was that hedging did not only occur within the advisory discursive moves (saving the face of the advisee) but also within assessments (saving the face of the advisor).

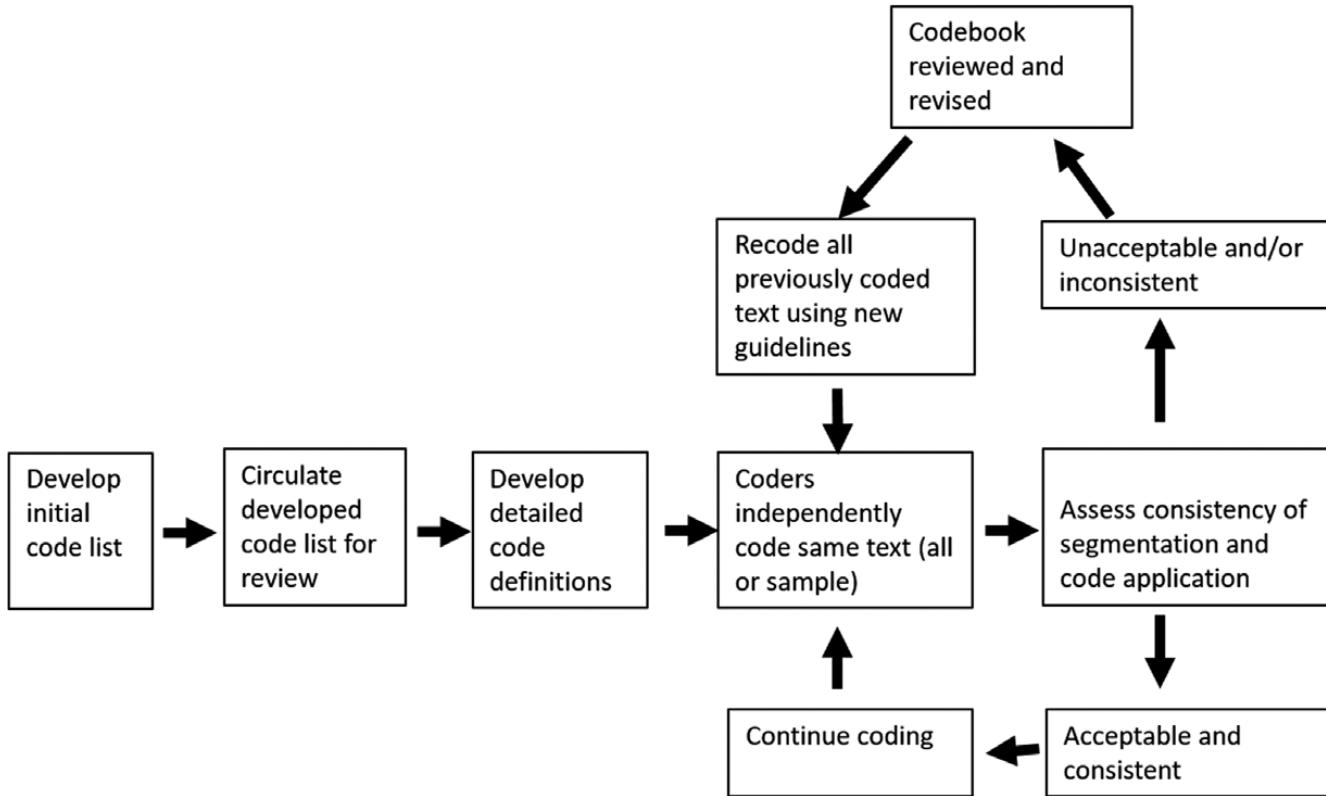
In Step 4, widening the scope, a number of further corpus linguistic and non-corpus linguistic qualitative steps were undertaken. For example, interviews were conducted with the professional health team, a computer-assisted diachronic study was conducted that compared original and updated entries to see whether just biomedical content or also the style of the entries changed, and the creation of the advisor voice was explored with case studies.

This scaffolded analysis resulted in a nesting structure that allowed the researchers to ask which discursive moves (Step 2) contained particular relational work strategies (Step 3), and to explore whether these realisations were topic sensitive (Step 1). This qualitative analysis was done for 10 per cent of the overall corpus, i.e. 280 question–answer letters, and made it possible to make robust statements about the patterns of the overall corpus. The codebooks, with definitions and illustrations of the codes, are included in the study and reliability of coding was achieved by testing the codebook with a group of peers.

The time it takes to establish a codebook can be substantial and needs to be taken into account in one's study design. Depending on your research question and approach, codebooks can draw on existing lists of codes (top-down), be developed from the data (bottom-up) or combine both possibilities. In all cases, reliability of codes and coding needs to be established. [MacQueen et al. \(2008\)](#) describe the cyclical development of a reliable codebook with a flow chart that shows how stable categories are developed before the entire dataset is analysed (see [Figure 1](#)).

Both [Rudolf von Rohr \(2018\)](#) and [Thurnherr \(2022\)](#) adopted and adapted these research steps to give justice to their data. They worked with the qualitative coding software *NVivo*. This software allows the coding of the same text passage with multiple codes and exploring co-coding (i.e. the layering of analysis). It also offers a function that establishes coder-agreement and has query options for cluster analysis of the codes and word frequency analysis, etc. Other commonly used qualitative coding software are for example MAXQDA and ATLAS.ti.

Rudolf von Rohr and Thurnherr first established what their texts are about in a qualitative analysis, secured with coder-agreement (Step 1, content/thematic analysis). The smoking cessation websites had different parts and sub-sites (e.g. facing quitting; inform on quitting; point out biomedical or lifestyle reasons to quit smoking; support) and the analysis was able to demonstrate the heterogeneity of the dataset. The therapy email exchanges were classified according to the problems that were raised by the counselee and detected by the therapist (e.g. anxiety or depression).



**Figure 1** Codebook development (MacQueen et al. 2008: 128)

In Step 2, they developed their own set of discursive moves bottom-up to qualitatively analyse the composition of the texts. In the case of Rudolf von Rohr, this was done for a subcorpus of websites and peer-to-peer support fora; in the case of Thurnherr, for the entire counselling exchanges. The development of a codebook was established bottom-up and was thus data-driven. This was an important desideratum so as not to impose pre-existing notions on the texts. However, as both practices have an advisory core, similar discursive moves emerged. Consider [Table 3](#), which shows the discursive moves for [Locher \(2006\)](#), [Rudolf von Rohr \(2018\)](#) and [Thurnherr \(2022\)](#). Comparable discursive moves appear in the same row. The table also shows that there are discursive moves unique to particular datasets, such the ‘Official forum welcome’ for the fora, or the ‘Introductory message’ written by the counsellor in the email therapy exchanges. Some of the discursive moves are only used by some interactors (e.g. problem letter writers (PL) versus response letter writers (RL) in *Lucy Answers*, thread initiators (I) vs post-respondents (R) in the forum data, or client (C) vs therapist (T) in the email therapy corpus). The discursive move analysis in combination with Step 1 thus allows scholars to find nuanced similarities and differences between comparable datasets. One of our findings is that the number of discursive moves to describe the practices in question is actually rather small within this advisory field, a result also confirmed by other studies that adopted a similar approach with discursive moves (see, e.g., [DeCapua & Dunham 2012](#); [Morrow 2012](#); [Placencia 2012](#)).

Looking at the realisation of relational work (Step 3), we found that it does not only matter how the advisory passages are linguistically realised (e.g. by exploring linguistic mitigation devices) but also how these passages are embedded within the overall composition of the texts, often resulting in mitigating effects.

In Step 4, widening the scope, [Rudolf von Rohr \(2018: ch. 6\)](#) explored how particular datasets clustered according to their use of discursive moves, drawing on the NVivo query options. Since her thirty websites differed in their multimodal composition, the computer-assisted analysis helped to find patterns. [Thurnherr \(2022\)](#) interviewed the counsellor who provided her data and organised workshops with counsellors to discuss results and to learn from the insights of the counsellors about their chosen strategies. She also drew on corpus linguistics tools such as word frequency and collocation analyses to explore the composition of discursive moves.

### 4.3 Conclusion

In all three studies of the *Language and Health Online* project, the computer-assisted qualitative data analysis software (CAQDAS) was crucial for the

**Table 3** Types of discursive moves used to analyse *Lucy Answers* (Locher 2006), smoking cessation website modules and fora (Rudolf von Rohr 2018) and email counselling (Thurnherr 2022) in alphabetical order

<i>Lucy Answers</i> PL = problem letter; RL = response letter	<b>Website modules</b>	<b>Email counselling</b> <b>I = thread initiator;</b> <b>R = respondent</b>	<b>Forum interaction</b> <b>C = client;</b> <b>T = therapist</b>
Advice (RL)	Advice	Advice (I, R)	Advice-giving (T)
Apology (PL)		Apology (I, R)	Apology (C, T)
Assessment (RL)	Assessment	Assessment (I, R)	Assessment (C, T)
Background information (PL)		Background information (I)	
Comment on previous record (PL)			
Compliment (PL)			
Disclaimer (RL)			
Explanation (PL, RL)	Explanation		
Farewell (RL)		Farewell (I, R)	Farewell (C, T)
General information (RL)	General information		General information (T)
Address (PL, RL)		Greeting (I, R)	Greeting (C, T)
	Header		Introductory message (T)
Metacomment (PL, RL)	Metacomment		Metacomment (C, T)

	Prediction Other voice	Official forum welcome (R)	
Own experience (RL) Problem statement (PL)		Own experience (I) Quote (I, R)	Problem statement (C) Quoting (C, T)
Question (PL) Referral (RL) Request advice (PL)		Request advice or information (I, R)	Referral (T) Request for advice (C)
Thanks (PL)		Thanks (I, R) Welcoming (I, R) Well-wishing (I, R)	Request for information (T) Scheduling (C, T) Thanks (C, T)

---

---

analysis (in our case *Oxygen XML Editor* and *NVivo*). It allowed the researchers to quantify and keep track of the qualitative analytic steps and, importantly, to explore the results of the scaffolded design of the methodology. A big advantage of qualitative coding software such as *ATLAS.ti*, *CATMA*, *MAXQDA* or *NVivo* is therefore that multiple coding of the same string is possible and that clustering can be explored. In addition, software such as *AntConc*, *WordSmith Tools* or *SPSS* can complement the corpus analysis with frequency, keywords and collocation analyses.

The analysis is informed by results from previous work as well as by the data itself. Developing the codebooks in pilot studies not only allows the establishment of coder-agreement but it also gives justice to the data since codes can be detected that are novel and unique to a particular dataset. The persistence of the analytic coding in the corpus also allows explorations that were not originally on the research agenda. For example, complementary to our own research interests, we explored a comparative perspective upon invitation. In Thurnherr et al. (2016), we explored our individual corpora top-down in order to explore narrative passages in our different datasets, and in Rudolf von Rohr and Locher (2020), we pursued the use of complimenting. In both cases, our bottom-up analysis of discursive moves had not yielded these particular concerns as primary, i.e. we did not create a discursive move for these units on their own (but complimenting was included as a relational work category in Locher 2006); however, we were able to reinterpret our previous analysis with the help of lexical searches of the corpora and then link the results to the discursive moves in which narrative passages and compliments occur. In other words, previous analysis can be combined with the exploration of new research interests.

The analyses discussed here in the context of the *Language and Health Online* project are fundamentally based on qualitative interpretation of lexical strings in context, but this process is quantified. The benefits of the quantification and scaffolding lie in being able to establish reproducible patterns that are representative for the corpus in question. It should not be denied that much of the time invested in this kind of exploration goes into establishing coder-agreement (MacQueen et al. 2008), which, however, is worth it in order to arrive at robust results.

Our brief introduction to the *Language and Health Online* project just described one of many possible ways to work with a custom-made corpus to gauge pragmatic research questions by drawing on and combining corpus linguistic methodologies and the possibilities of layered annotation. Other corpus-assisted studies (see Section 4.1 for a selection) make use of different corpus linguistics tools in order to do pragmatic analyses. Key to all of these endeavours is that the corpus-assisted approach to pragmatics aims at

combining qualitative with quantitative steps of analysis. Referring to a corpus in a replicable and systematic way enables the creation of taxonomies, ensuring their descriptive adequacy and allowing us to validate them empirically.

## 5 Compatibility and Comparability: Combining Existing Corpora

### 5.1 Introduction

Working with a combination of several corpora can take different forms. In many cases, researchers access each corpus individually and compare results, for instance on the basis of normalised frequencies. In other cases, existing corpus resources are combined into a new corpus. There are many reasons why researchers may want to work with several corpora at once. For instance, they may be interested in comparing different varieties of a language or different languages by using a corpus for each language or language variety. Perhaps they want to carry out a diachronic study by comparing corpora from different periods. They may want to make comparisons across different domains by working with several domain-specific corpora. Or they may simply want to base their observations on a larger set of data by combining different corpora.

This last point is particularly relevant for historical studies (see also [Brinton 2012](#)). Whereas researchers interested in present-day language usually have the option of expanding the data that is available to them by collecting more data, this is often not the case for historical studies. Here, research is limited to texts that have survived and, at least for the earliest periods of English, most of these have already been included into the corpora we have available today. For instance, the *Dictionary of Old English Corpus* (*DOE*) includes almost all surviving texts from the Old English period, amounting to roughly 3 million words of Old English. For slightly later periods of English, such as Late Middle English and Early Modern English, material not included in existing corpora can still be found in archives, but access restrictions to such material and the nature of the data mean that transforming it into electronic corpus data is usually an extremely time-intensive process. As a consequence, it often takes entire teams of researchers working on such corpora for several years before the data is ready to work with. Thus, instead of compiling a new, larger corpus, it makes sense to build on the work that has already been carried out by working with several existing corpora in combination.

However, working with several corpora side-by-side brings with it a range of challenges. These challenges can be summarised under two different headings: compatibility and comparability. With compatibility we refer to formal and technical aspects, ranging from the way in which corpus data is edited and stored, how it can be accessed and searched, and what metainformation is

available. These aspects influence how researchers can work with different corpora side-by-side and how easy or difficult it is to retrieve the same kind of information from different corpora – or whether this is in fact impossible. For instance, if one of two corpora includes syntactic annotation, but the other does not, comparisons across these corpora that rely on syntactic annotation cannot be made. With comparability, we refer to differences in the data that are included in the corpora, as well as to differences in the calculation methods of corpus tools. While compatibility issues affect whether or not information can be retrieved from all corpora used in a given study, comparability issues affect whether or not the retrieved information can be compared across corpora in a meaningful way. For instance, if two corpora include syntactic annotation, but the way in which syntactic units were classified differs across the two corpora, then the results may not be comparable, because they are influenced by differences in the annotation schemes. Researchers need to know their corpora extremely well in order to be aware of differences across the corpus data that affect their results and interpretations. Compatibility and comparability of corpora are challenges that all corpus linguists face when they work with different corpora. They are not restricted to corpus pragmatics, but, of course, they need to be addressed when studying pragmatic features with corpora.

In the remainder of this section, we first discuss issues of compatibility ([Section 5.2](#)) and comparability ([Section 5.3](#)). In [Section 5.4](#), we then present a study that is based on a combination of different Early Modern English corpora to illustrate some of these issues in more detail.

## 5.2 Compatibility

When starting to work with different corpora, practical problems often appear quite early on. Unless the corpora are provided by the same research team or included on the same platform, even gaining access to the corpus data may take very different forms. Some corpora are available for full-text downloads, some can be copied from CD-ROMs and others can only be accessed through online search interfaces. This last option involves most restrictions. Researchers have to rely on the corpus tools that are provided by the online interface, and they cannot reprocess data, for instance by applying their own part-of-speech tagging. If different corpora have to be accessed through different search interfaces, it is possible that different tools are offered on the two sites, or – perhaps even worse – that the tools appear to be the same, but that they apply different methods. Unfortunately, not all corpus interfaces document in a clear and transparent way how they operate, and differences in calculation methods can have profound effects on comparability (see [Section 5.3](#)).

For corpora that are available as full-text versions, gaining an overview of the corpus files, their format and how they can be processed is sometimes not straightforward. There are corpora that come as collections of folders and files with an intuitive structure, such as folders and files per text type or period. Others, like the downloadable version of the *Movie Corpus*, are presented in the form of huge text files with no apparent order of the data and only minimal metainformation in the form of a numerical ID within the text files, which has to be cross-checked in an Excel file. While such information can be processed by computers, this form of presentation makes it harder for researchers to understand the structure of the data. In addition, many corpora are available in different versions: ordered by text type or period (e.g. *ARCHER*), as text only, part-of-speech tagged or parsed (e.g. *Parsed Corpus of Early English Correspondence (PCEEC)*), with or without spelling normalisation (e.g. *Early Modern English Medical Texts (EMEMT)*), or with or without annotation (*CED*). Researchers need to know a great deal about all the corpora they want to work with in order to choose the corpus version that best fits their needs.

Perhaps the biggest practical problem for downloadable corpora is posed by different file formats. Nowadays, the most common format by far is XML, but there are still corpora that are available in other formats. Especially for older corpora, text files with custom-made annotation formats are not uncommon. Even if all corpora are available in XML format, different corpora often use different conventions with respect to the tagset and presentation of metainformation. The Text Encoding Initiative (TEI) has developed extensive guidelines and recommendations on how to present various kinds of text material in digital form based on XML. However, while these conventions have been applied in some corpus projects, many compilers still decide to develop their own annotation scheme, custom-tailored towards their corpus data and the needs of their study. When working with corpora in different formats, researchers need to decide whether or not to transform all of them into one single format. This process can be time-consuming, and it can also involve the loss of metainformation. The advantage is, however, that the corpora can then be processed in the same way, which makes the analysis easier and improves the comparability of the results.

One way of dealing with compatibility problems is by removing most metainformation from a corpus and treating it as a collection of text material that can be further processed with a given set of corpus tools. This is what multi-corpus tools like the commercially available *Sketch Engine* often do. *Sketch Engine* is an online interface that provides access to the content of 600 corpora in more than ninety languages. While this means that researchers can gain access to a dazzling amount of data, much of the metainformation of the individual corpora is lost and the access to the corpus texts is strictly limited by the search interface.

This poses serious problems for the interpretation of results. Especially for studies that rely on qualitative interpretations of retrieved instances, such platforms are of very limited use. For pragmatic studies, for which access to contextual information is crucial, this approach is usually not suitable.

Even when corpora are prepared by researchers themselves, the goal of achieving compatibility without sacrificing richness of information is often not easy to achieve. Different types of data sometimes simply require different types of formatting and metainformation. For instance, for a corpus of drama texts, it is quite important to be able to distinguish between speaker labels, lines that are spoken by characters and stage directions. In contrast, such annotations make little sense for a corpus of newspaper data. However, in the latter, it may be important to distinguish between up-, mid- and down-market newspapers, between hard news articles and opinion pieces, and between headlines and the main body of an article (see, for instance, [Landert 2014](#)). For research questions for which such annotations play a role, decisions need to be taken with respect to how much annotation is retained. Retaining annotations may provide additional information for the interpretation of pragmatic functions and distributions, while at the same time making the datasets less compatible.

While this section has illustrated challenges with respect to compatibility of the formal and technical aspects of the corpora, [Section 5.3](#) illustrates the challenges encountered with respect to comparability, i.e. whether the data that is included in and retrieved from the corpora is comparable.

### 5.3 Comparability

Comparability becomes an issue at two different levels: the level of the data that is included in a corpus (e.g. which registers, period, speakers, etc.) and the methods that are used to process the corpus data (e.g. how frequencies and collocations are calculated). The first of these issues is perhaps the more obvious one. For instance, if we want to make a comparison across two language varieties – e.g. British English and American English – by using a corpus for each variety, we need to be sure that the corpora are as similar to each other as possible in terms of corpus composition. This means that they should include the same proportion of texts from given registers, text types and language users. If this is not the case, any difference in corpus composition may distort the results. While this is an obvious point, it poses many practical problems, given that close similarity of composition is often not given.

There are various corpus projects that try to maximise comparability of corpus data across different varieties and/or time periods. Examples are the *Brown* and *Lancaster–Oslo/Bergen (LOB)* family of corpora, which consist of

a set of corpora of American English (*Brown*) and British English (*LOB*). The original corpora, which included data from 1961, were complemented with comparable corpora from 1991/1992. Additional corpus components from later and earlier periods are published, and others are currently being compiled. For spoken language, the *BNClab* platform provides access to two subsets from the original *BNC1994* and the new *BNC2014*, which include demographically sampled data that can be used for sociolinguistic studies of language change. Another project that tries to maximise comparability of corpus data across different varieties of English is the *ICE* (see also [Section 2.2](#)). It consists of twenty-six corpora – some completed, some still in progress – each devoted to a different variety of English. Each corpus is compiled according to the same design principles, e.g. including 300 text samples of spoken and 200 samples of written language, which are spread across different genres (such as classroom lessons, broadcast discussions, parliamentary debates) in predefined ways (see [www.ice-corpora.uzh.ch/en/design.html](http://www.ice-corpora.uzh.ch/en/design.html)). However, in practical terms, some differences in corpus composition still occur even between these corpora. Some differences are related to the fact that some types of texts are not produced or not accessible in a given variety (e.g. parliamentary debates in the English of Malta, see [Hilbert and Krug 2010](#): 60). Other differences are introduced through the duration of the project: while the British component of the corpus, *ICE-GB*, was compiled in the 1990s, new components are still in the process of being compiled, which means that data is collected about twenty-five years later than for the first corpus. Thus, with respect to time period, the corpora are not perfectly comparable.

In historical corpora, the kind of precision of comparability that the *ICE* offers can hardly be achieved. There are very practical limitations to corpus composition in the form of lack of surviving and accessible data. Moreover, text types and genres change over time, meaning that diachronic corpora spanning a long period cannot have exactly the same structural composition (see [Taavitsainen 2016](#)). For instance, in the *Helsinki Corpus*, ‘drama comedy’, ‘trial proceedings’ and ‘personal diaries’ are not included for Old and Middle English, because they either did not exist in, did not survive or were not accessible from these periods. In addition, text types and genres undergo development, too, so that texts from the same genre can include different linguistic characteristics at different points in time not because language overall has changed, but as a reflection of the development of the genre. For instance, [Kohnen \(1997\)](#) showed in an early study of the *Helsinki Corpus* that the increase of gerund constructions with direct object in the corpus section ‘fiction’ can be explained by the development of narrative prose, which resulted in longer and more complex fictional texts over time. Thus, the text category

‘fiction’ in the fifteenth century is quite different from the category ‘fiction’ in the seventeenth century. Given the context-dependency of pragmatic meaning, changes in text types and genres are likely to affect pragmatic forms and functions at least as much as lexical and grammatical characteristics.

Such differences in genre development and corpus composition across periods can have considerable effects on the results of linguistic studies, and they need to be taken into account for the interpretation. More problematic than known differences in corpus composition, therefore, are differences of which researchers are not aware. Familiarising oneself with the content and structure of a corpus is one of the most important steps in any corpus-pragmatic study.

When working with multiple corpora, it is highly advisable to either work with the full-text version of each corpus locally or to access them through the same interface – assuming that all corpora were processed identically on a given platform. If one has to rely on different corpus tools for different sets of data, there is always a danger that differences in the tools lead to different results, due to differences in calculation methods. To give an example, word counts can be calculated in many different ways: treating hyphenated words as one or two words, counting numbers as words or not, or whether or not to include punctuation in the word count. Such differences in how word counts are calculated can distort normalised frequencies. Frequency normalisation is a crucial step when working with several corpora. All results need to be presented in relation to the overall size of the corpus, which happens by dividing the number of instances retrieved from a corpus by the total number of words in the corpus. If the way in which word counts are calculated differs across different corpora, then frequency normalisation is affected and the normalised frequencies are no longer comparable.

In order to illustrate the variation in word count, [Table 4](#) gives an overview of eight different results for the overall number of words in the *Sydney Corpus of Television Series*, four each for the non-standardised (*SydTV*) and partly standardised (*SydTV-Std*) version of the corpus. These results are provided by

**Table 4** Number of words in the *Sydney Corpus of Television Dialogue*, according to corpus documentation ([Bednarek 2018](#): 253)

<i>Token definitions (Wordsmith ‘tokens in text’)</i>	<i>SydTV</i>	<i>SydTV-Std</i>
hyphens do not separate words; ‘ not allowed within word	275,074	276,899
hyphens separate words; ‘ not allowed within words	276,287	278,112
hyphens do not separate words; ‘ allowed within word	258,944	260,824
hyphens separate words; ‘ allowed within word	260,157	262,037

**Table 5** Word counts of the *Sydney Corpus of Television Dialogue* on *CQPweb* interface

	<i>SydTV</i>	<i>SydTV-Std</i>
Sydney Corpus of Television Dialogue	334,379	335,087
Sydney Corpus of television Dialogue CLAWS	334,998	335,223

Bednarek (2018: 253), who compiled the corpus, and they include an unusual amount of detailed information with respect to how they were calculated.

As these numbers show, even within the same corpus version, word counts can vary for around 5 per cent depending on how hyphens and apostrophes are treated. In addition to these differences, Table 5 shows four additional results for the word counts of the same corpus as they are provided on *CQPweb*, the interface through which the corpus can be accessed. Again, there are some differences between the version using the CLAWS tagset and the other version. More importantly, though, the numbers differ from those given by Bednarek (2018) by about 20 per cent. According to Bednarek (personal communication), this difference is due to differences in the treatment of multi-word units and punctuation.

The point here is not to assess which method of establishing word counts is the correct or the best method, but rather to raise awareness that such differences in establishing word counts can seriously distort results when comparing results derived from datasets that use different tokenisation methods. As a consequence, it is highly advisable to use the same methods and tools on all sets of data.

#### 5.4 Case Study: Combining Four Early Modern English Corpora

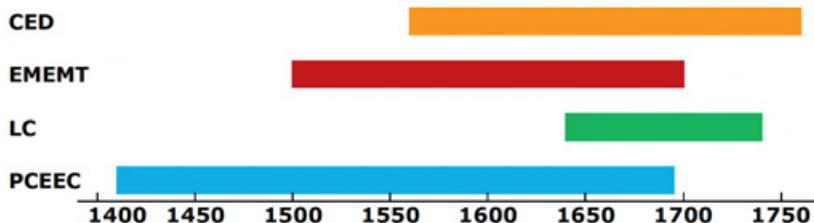
This section illustrates some of the issues that arise when working with several corpora side-by-side by referring to a research project on epistemic stance in Early Modern English (Landert 2019). The main research of the project was based on four existing corpora: the *Corpus of English Dialogues 1560–1760* (*CED*), the *Early Modern English Medical Texts* corpus (*EMEMT*), the *Lampeter Corpus of Early English Tracts* (*LC*) and the *Parsed Corpus of Early English Correspondence* (*PCEEC*). These four topic- and domain-specific corpora are mid-sized, ranging from 1.2 to 2.2 million words, and all four corpora had been compiled with a great deal of care and extensive documentation with the aim of making it possible for other researchers to use them. Despite these ideal conditions, working with the corpora was not without problems.

Several of the corpora were available in more than one version, which made it necessary to decide which corpus version to use. For instance, the *EMEMT*

was available in a version with normalised spelling and in a non-normalised version. For many parts of the analysis, using the normalised version would have been an advantage, since it would have removed the need to take spelling variation into account when formulating search terms. However, given that not all the corpora were available with normalised spelling, the non-normalised version was chosen in order to maintain comparability of the results. For the *PCEEC*, the corpus release that was available was based on the parsed version of the corpus, but for the project, the text-only version was used. While several of the corpora included versions with part-of-speech tagging, it was not clear to what extent the tagging was comparable between the corpora and, therefore, the text-only version without any linguistic tagging was used for all four corpora. For the study in question, no part-of-speech tagging was applied.

One aspect in which it was not possible to make the corpora comparable was time span. While all four corpora included Early Modern English data, they differed quite considerably with respect to the period they covered (see Figure 2). While the *PCEEC* starts in 1410, including some material from Late Middle English, the *LC* starts only in 1640, overlapping with the *PCEEC* for merely fifty-five years. Reducing the material to just the time span that is covered by all four corpora would have restricted the data too much. As a consequence, the entire corpora were included in the analysis, but the difference in coverage was considered during the interpretation.

Compatibility of corpus formats provided the biggest practical obstacle by far. Only one of the corpora, the *CED*, was available in XML format. This corpus was kept in XML, but some aspects of the annotation scheme were adjusted in order to facilitate later steps in the analysis. For instance, the name of the document-level tag was changed from <*diaolgueDoc*> to <*document*>, which was a tag that could be used for all four corpora. In this way, the same XPath query could be used to search all corpora. In addition, some



**Figure 2** Time span of four Early Modern English corpora (*CED* = *Corpus of English Dialogues 1560–1760*; *EMEMT* = *Early Modern English Medical Texts* corpus; *LC* = *Lampeter Corpus of Early English Tracts*; *PCEEC* = *Parsed Corpus of Early English Correspondence*)

metainformation was integrated into the <document> tag in the form of attributes, such as the corpus section and the period of the text. Encoding this information as attribute made it possible to use it as filter criteria of XPath queries when searching the corpus.

The other three corpora had to be transformed to XML first. The *EMEMT* was transformed from an ASCII text version, the *PCEEC* from an HTML version and the *LC* from an SGML version – a format that was very innovative at the time the corpus was compiled but which proved to require quite some manual work for the transformation.

Irrespective of the format, the kind of information that was annotated differed considerably, due in part to differences in the nature of the data. To the extent that they described aspects of the data that were relevant for the analysis, many of these differences were retained, such as the distinction between speech representation and non-speech representation in the *CED*. Some types of information were considered not to be central for the scope of the project and these were removed whenever they would have introduced additional complexity into the data. For instance, the *LC* included very detailed information about the layout of the printed tracts, which was not taken into account in the analysis.

Following the transformation of the corpora to compatible XML versions, the four corpora were processed with the same tools, scripts and methods. In this way, the comparability of the results from the corpora was maximised. Potential effects on the results that were related to the corpus composition, the nature of the historical data, and genre- and text-type specific factors were considered throughout the interpretation. Some information about subsequent steps of the analysis are given in [Section 6.4](#).

## 5.5 Conclusion

Combining different corpora in a single study can greatly enhance the available data and it can also open up research perspectives that could not be pursued with any single corpus on its own. At the same time, working with resources that were compiled at different times by different researchers with different research questions in mind and which can sometimes only be accessed by using different interfaces and tools creates a range of challenges. While any researcher who works with a corpus should know their corpus well, this is probably even more true for researchers who work with several different corpora in combination. By familiarising oneself very well with a corpus, its composition, its formatting and the tools that may come along with it, many problems that may affect the results can be identified and, ideally, avoided.

## 6 Scalability: Meaningful Pragmatics with Large Data

### 6.1 Processing Time and Corpus Size

Scalability as we use it in this section, refers to the ease with which a given method can be transferred to other, especially larger, corpora. Given the overall tendency of corpora to increase in size, it is worthwhile to discuss scalability and its consequences for quantitative and qualitative corpus methods. The factor that affects scalability most of all is the question of how the processing time a method requires relates to corpus size. It is important to note that processing time here refers not only to the processing time of computers. It involves all the steps involved in arriving at the result, from accessing the corpus and formulating queries, to reviewing and analysing results. In practical terms, computer processing time is negligible nowadays for most of the typical corpus pragmatic methods. The processing power of computers has increased a great deal and, as a consequence, the automatic computation of results has become a smaller part of the overall process, as far as duration is concerned. Thus, the other steps in the process tend to decide how processing time increases in relation to corpus size.

For the majority of the quantitative corpus methods that are most often used in corpus pragmatics – such as normalised frequencies, frequent collocations, n-grams and keywords – the automatic computation of results is so quick that the overall time needed for the analysis hardly increases with corpus size. For instance, if we want to calculate normalised frequencies based on a huge corpus, all we need are the number of words in the corpus and the number of instances we are interested in. As long as these two numbers are returned automatically by the corpus tools, we have the results within seconds of entering our query, even if the query is run on a very large corpus. When applying restrictions on queries that go beyond searches for strings, such as part-of-speech tags or lemma searches, queries can become noticeably slower for large corpora. Still, in most cases, queries will return results in under a minute, often even in under ten seconds. For instance, the calculation examples provided by Davies on the 1.9 billion word *Corpus of Global Web-Based English (GloWbE)* corpus for queries including lemma searches and part-of-speech tagged elements all range from two to seven seconds ([www.english-corpora.org/speed.asp](http://www.english-corpora.org/speed.asp)). Thus, while the computer processing time of calculating the number of instances will increase with corpus size, the overall duration of the process is so short that it hardly matters in practical terms.

For other common corpus operations, such as for the calculation of collocations and keywords, the increase of computer processing time with corpus size may be more noticeable. For very large corpora, computer processing time can become an issue for such operations. However, in practical terms, corpus

infrastructures often offer solutions to deal with such issues, for instance by including pre-calculations of word frequencies that speed up the process.

However, the situation changes drastically when we turn to qualitative analysis. Methods that require researchers to analyse each result manually have very poor scalability for very large corpora. If the number of relevant instances increases with corpus size, then the amount of time to analyse these instances increases in a linear fashion as well. For rare linguistic phenomena, very large corpora are still an advantage, but even for moderately frequent phenomena, large corpora can soon produce more results than a researcher can analyse in a qualitative manner. This means that larger corpora are not always an advantage for such studies. Although they may include more potentially valuable material for qualitative analysis, finding it in the mass of data may become very difficult.

In the remainder of this section, we will discuss some old and new approaches to this challenge and we will discuss options for making the most out of large amounts of corpus data, even for studies that require a great deal of qualitative analysis. We will first discuss sampling, as a well-established way of dealing with large sets of retrieved hits, before turning to more recent approaches that aim at retrieving texts or passages of texts from a corpus that can provide particularly rich insight into a given pragmatic phenomenon.

## 6.2 Sampling

A popular approach to limit the number of person-hours spent on manually annotating pragmatic phenomena is to use a small section of the entire corpus, i.e. a sample. The assumption behind sampling is that the sample is sufficiently representative of the entire corpus for the researcher to be able to generalise the findings to the corpus, and consequently, to the larger population. It is of crucial importance to base sampling on systematic decisions rather than convenience in order to make sure that the sample includes the full range of variability in the population (Biber 1993). These decisions will depend on the type of corpus in question and always need to be carefully documented.

The same sampling principles apply to choosing a subsample from our corpus as from the population at large: in corpus linguistics, researchers traditionally resort either to simple random sampling or to stratified sampling. To conduct random sampling, the analyst numbers the sampling units in the collection (most commonly, individual corpus files in the corpus) and chooses the sample using a random numbers generator. The drawback of this method is that the resulting sample might not include relatively rare items in the corpus, since the chance of an item being chosen correlates with its frequency in the corpus (McEnery et al. 2005).

Stratified sampling can alleviate this problem by sampling from various parts of the corpus in a balanced way. When sampling from a population, we first divide it into homogenous groups, or strata, according to a chosen principle: for example, demographic strata such as age and gender of the speaker. Data from each stratum is then sampled randomly. When drawing a subsample from a corpus, it is of course possible to rely on some aspects of the pre-existing sampling frame as strata. For instance, [Maynard and Leicher \(2007: 111\)](#) created a subsample of the *Michigan Corpus of Academic Spoken English (MICASE)* corpus for their pragmatic annotation project by selecting fifty transcripts from the total of 152 transcripts based on the original sampling category ‘academic division’ (humanities and arts, physical sciences and engineering, etc.).

One limitation of choosing the subsample for manual pragmatic annotation in this manner is that the sampling frame of a corpus may not fit the present research question. In that case, the researcher may choose a different stratification principle that would result in a more adequate sample. In pragmatic research, meaningful strata are often thematic, although structural criteria such as the number of discourse participants, the length of corpus file or the time of production are also used. An instance of this process is the study of illocutionary acts in SMS by [Sotillo \(2012\)](#), who sampled 1,217 text messages from her overall corpus of 5,809 messages by dividing the subsets of data donated by each study participant into three groups chronologically ('beginning', 'middle' and 'end' of the 1.5 year period of data collection) and then randomly sampling from these three categories.

Finally, two considerations to keep in mind concern the sampling unit and sample size. These should be based on the careful consideration of the research question, although convenience may drive the decision to keep the pre-existing sampling unit and to minimise the size of the sample to be annotated. While many corpora are built around the unit of an entire text or a speech event, a more meaningful choice for a pragmatic study might be to sample only the beginnings of texts or only the turns by one interlocutor (as, for instance, [Weisser \(2018: 271\)](#) observed that greetings occur predominantly in the beginnings and ends of the Switchboard corpus files). The choice of the sampling unit and the sampling process itself can indeed become a powerful tool in optimising the annotation efficiency, as [Sections 6.3](#) and [6.4](#) will demonstrate.

### **6.3 Case Study: Manual Identification of High-Density Corpus Files for the Study of Self-Praise**

This section addresses the speech act of self-praise: an expressive speech act that gives credit to the speaker for some attribute which is positively valued by

the speaker and the potential audience (Dayter 2016: 65). While linguistic approaches to politeness prior to the discursive turn described self-praise as face-threatening and inappropriate, recently it has been recognised as an integral part of communication across a wide variety of communicative contexts (Matley 2017; Dayter 2018).

‘Self-praise’ and associated metapragmatic labels such as ‘bragging’ or ‘boasting’ are negatively charged and seldom used. This speech act, which also lacks stable IFIDs and metacommunicative expressions, is difficult to extract from large corpora using any of the three methods mentioned in Section 1.4.<sup>2</sup> At the same time, the empirical study of self-praise using corpus approaches remains crucial for the adequate description of this long-neglected speech act.

The first steps towards identifying self-praise in a corpus used fully manual annotation (Dayter 2016) and semi-automatic annotation through a list of candidate IFIDs (Dayter 2018). However, such approaches are not easily scalable to large datasets. An alternative approach to identifying this pragmatic phenomenon, aiming to be scalable, is described in Dayter (2021). The driving assumption behind the method is that even in the corpora that are relatively homogenous in terms of genre, some text excerpts may be richer in the pragmatic phenomenon in question than others. To give a somewhat trivial example, if we are interested in a study of greetings, it would be efficient to limit our manual annotation to the first few turns of every conversation included in a corpus of spoken interactions. Although such an approach might miss some untypical instances of greetings (for example, when an interlocutor joins a conversation well underway), it will nevertheless achieve high sensitivity and save the human annotator the significant effort of closely reading the remaining bulk of corpus texts.

The speech act of self-praise does not immediately offer such a convenient springboard to limit the search. It is, however, conceivable that self-praise might occur more frequently in some subtypes of speech events than in others. In order to test this hypothesis, Dayter (2021) conducted a pilot annotation of her domain-specific 230,000 words corpus. The corpus included seventy-two speech events in high-level political contexts that were predominantly monologic and delivered in the one-to-many modality: addresses at the United Nations (UN) General Assembly, UN Security Council, UN Human Rights Council and UN Universal Periodic Review sessions, as well as statement sections of presidential and ministerial press-conferences. The study established that self-praise by politicians overwhelmingly occurs in a certain kind of oral report. This is a specific United Nations rhetorical subgenre (the

<sup>2</sup> The three methods are (1) the search for IFIDs, (2) the search for known lexicogrammatical patterns and (3) the study of metacommunicative expressions (see Section 1.4).

statement section of the Universal Periodic Review (UPR) of the human rights records of the UN member states) which calls for the speaker to report on their latest achievements. Although other speech events in the corpus similarly offer speakers an opportunity to present their activities in a positive light, UPR reports emerged as the locus of self-praise.

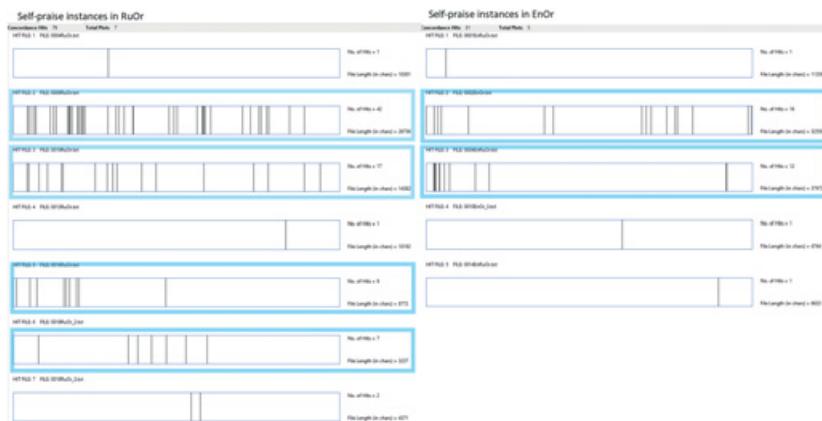
In order to catch as many instances of self-praise as possible, annotation was conducted in two rounds. The first round relied on semi-automatic identification of self-praise excerpts through search queries using candidate IFIDs (such as *the best*, *the most*, *leading*) and metapragmatic commentary (such as *praise*). The retrieved concordance lines were manually checked to confirm the presence of self-praise.

Although combining IFIDs with metapragmatic remarks is a fairly successful approach in detecting explicit and/or direct self-praise (Example 6.1), this speech act is often performed implicitly and indirectly (Example 6.2, which is impossible to recognise as self-praise without the detailed understanding of the co-text), or directly, but using creative lexical items (Example 6.3).

- (6.1) Belarus has the best indicator in the CIS region in the level of child mortality (0009EnTr)
- (6.2) we have also submitted an interim report on the review of implementing the recommendations made at the end of the first UPR (0009EnTr)
- (6.3) I believe America is exceptional in part because we've shown a willingness through this sacrifice of blood and treasure to stand up not only for own narrow self-interest but also for the interest of all (0002RuTr)

Consequently, every speech event where self-praise had been detected in the first round of extraction was manually annotated to catch the instances of self-praise not marked by IFIDs or metapragmatic expressions. After reviewing the distribution of self-praise across the corpus files, three files emerged as especially dense in self-praise. All of these belonged to the category of statements within the Universal Periodic Review. All in all, twelve transcripts of speech events out of the total seventy-two events in the corpus contained self-praise. Six of these belong to the category of UPR statements and, as Figure 3 demonstrates, are much denser in self-praise (highlighted) than other subgenres.

To sum up, further self-praise research on UN political discourse involving manual annotation of the highly interpretative category of self-praise (for example, following the corpus-assisted methodology outlined in Section 4 of this Element) can be conducted more efficiently if the annotators' efforts are limited to the UPR statements. Although this example concerns a relatively small spoken corpus, Section 6.4 describes the successful adoption of the principle to



**Figure 3** Concordance plot of self-praise distribution in the corpus of political discourse, adapted from Dayter (2021: 34), produced in AntConc 3.5.0. UPR statements are highlighted with frames (RuOr: Russian Original; EnOr: English Original).

automatically identify high-density passages in a much larger corpus and thus scale the study of a pragmatic phenomenon to a corpus of many million words.

#### 6.4 Case Study: Automatic Extraction of High-Density Passages of Epistemic Stance in Early Modern English

Pragmatic phenomena tend to cluster not just in certain text documents but even in certain passages of documents. For instance, for requests, Culpeper and Archer (2008) identify a tendency of using multiple requests in the same turn for their data from Early Modern English trials and, to a lesser extent, in drama. Likewise, Archer and Gillings (2020), in their study of lying and deception in Shakespeare's plays, note that deceptive features tend to occur in clusters. Vaughan et al. (2017) find that certain items of vague language tend to co-occur. And, as Andersen (2011: 601) points out, studies have repeatedly found that discourse markers tend to cluster together with other discourse markers (see also Aijmer 2013). So far, this tendency of pragmatic features to cluster together has often been identified as a result of linguistic studies. Only rarely have clusters been taken as the starting point of linguistic methods. By developing methods that identify clusters of related pragmatic features in texts, we can complement existing approaches.

For instance, if we know that pragmatic features have a tendency to cluster together, we can use this fact in order to develop analytical steps that allow us to identify and extract passages in corpora that contain such

**Table 6** Overview of the five analytical steps to identify stance markers in Landert (2019)

<b>Step 1</b>	Identification of frequent and reliable stance markers	Corpus-based, manual analysis of stratified sample
<b>Step 2</b>	Tagging of frequent and reliable stance markers in corpora	Automatic with Python script
<b>Step 3</b>	Identification of text passages with a high density of tagged items	Automatic with Python script
<b>Step 4</b>	Analysis of high-density passages and identification of new stance markers	Manual analysis
<b>Step 5</b>	Further analysis of newly identified markers	Corpus-based analysis

**Table 7** Twenty frequent and reliable stance markers in Early Modern English (Landert 2019: 177)

<b>Verbs</b>	<i>believe, doubt, know, perceive, seem, suppose, think</i>
<b>Adjectives</b>	<i>confident, evident, (un)likely, manifest, (im)possible, (im)probable, (un)sure</i>
<b>Adverbs</b>	<i>certainly, perhaps, plainly, surely, truly, verily</i>

clusters. Once found, these passages can then provide rich material for qualitative analysis. Thus, instead of analysing random or stratified subsets of individual instances from a corpus, extra work is invested in identifying passages that include such clusters, which are then submitted to qualitative analysis. The time-intensive manual investigation can be applied to those passages in a corpus that include most instances and which, as a consequence, may be especially relevant for understanding the phenomenon under investigation. In this way, more data can lead to richer observations without a linear increase of work time.

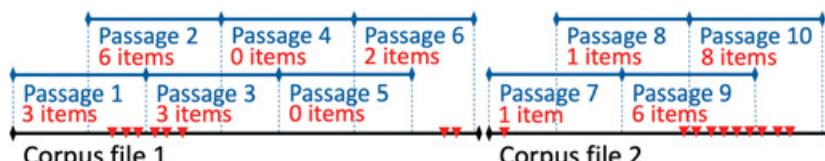
This idea has been applied to the study of epistemic stance in Early Modern English (Landert 2019). One of the aims of the research project was to identify stance expressions that had not been discussed in previous research. This included period-specific stance markers, lexical expressions that are only occasionally used as stance markers, and expressions that imply stance and whose stance meaning relies on an interpretation of the expression in context. In other

words, the study adopted a function-to-form approach in which the full inventory of the forms was not known beforehand.

The method that was used to identify and analyse stance markers consists of several steps (see [Table 6](#)). In Step 1, a small set of frequent and reliable stance markers was identified. These items served as seed items that helped identify passages with a high density of stance markers. The candidate items for this set of markers were in part identified based on findings from previous research. The list was restricted to frequent lexical items and a manual analysis of a stratified sample of each lexical item was used to exclude items that were often used with meanings not related to stance. The final list of frequent and reliable stance markers consisted of twenty lexical items (see [Table 7](#)).

In Step 2, all instances of the twenty frequent and reliable stance markers were automatically tagged in four Early Modern English corpora (see [Section 5.4](#) for a discussion of the four corpora). Following this, Step 3 involved the identification of passages that included many tagged items. Like Step 2, this was done in a fully automatic way, using a Python script. The study adopted a sliding-window approach, in which the number of tagged items was calculated in overlapping 300-word-passages (see [Figure 4](#)). The script then returned those passages with the highest number of tagged items. In Step 4, the passages with most tagged items were manually analysed and all stance markers in these passages were identified. The analysis showed that the passages that included many tagged items also contained other stance markers, including markers that had not been described so far. For instance, the verbs *collect* (meaning ‘conclude’) and *credit* (meaning ‘believe’) can both be used with epistemic meaning across all four Early Modern English corpora. Both verbs are not included in previous studies of epistemic stance. In Step 5, the final step of the analysis, the newly identified markers were studied in more detail in all four corpora, using established corpus-based methods (form-to-function approach).

The central step consists of the manual analysis of the high-density passages (Step 4). This detailed qualitative investigation can result in rich observations concerning the use of epistemic stance in Early Modern



**Figure 4** Illustration of the sliding-window approach to identify passages with a high density of tagged items

English. By retrieving passages that include a high density of known stance markers, the time-intensive manual analysis could be restricted to passages for which it was highly likely that they would result in relevant observations. In this way, the size of the combined corpora – about 6.6 million words – did not present an obstacle for the qualitative step in the analysis. On the contrary, more data means that it becomes more likely that passages with many stance markers are retrieved. By restricting the analysis to those passages with most tagged items, the number of passages that are analysed manually can remain manageable, even if more data is available overall.

## 6.5 Conclusion

In this section, we looked at ways in which studies that require a great deal of time-intensive qualitative analysis can still make use of the increasing size of corpus data that has become available. One key for resolving the tension between qualitative analysis and corpus size is the observation that many pragmatic features are not evenly distributed across texts. If we find ways of identifying text types, texts and passages within texts which include a high number of instances of the feature we are interested in, we can restrict the time-intensive qualitative analysis to these. Developing such methods will help improve the scalability of qualitative methods, so that they, too, can make the best use of the rich potential offered by large corpora.

# 7 Multimodality: Integrating Non-verbal Information

## 7.1 Introduction: Corpora beyond Language

When applied in corpus linguistics, multimodality broadly refers to the representation of more than one mode in a corpus. This may seem conceptually simple – understanding illustrations as well as written sentences as part of a book, or gaze and speech as part of a spoken interaction – but multimodality is in fact hard to define and operationalise for our context of corpus pragmatics. It is important to point out first of all that even monomodal corpora – consisting of systematically organised collections of written texts – are the product of a modal and/or medial transfer. Following the terminology employed, e.g. by Kress and Van Leeuwen (2001), Kaindl (2013) and Messerli (2020), corpus building may include an intramodal and intramedial transfer from written to written and from online text to online text. For instance, in the case of the *NOW* corpus, texts embedded in online environments such as news sites are recontextualised as online corpus texts read or searched on english-corpora.org.

Alternatively, as is the case for *CHILDES* in the TalkBank system, an inter-modal and intermedial transfer may take place from spoken to written and from face-to-face interaction to online text. While the *CHILDES* corpus consists primarily of transcripts, the data it represents consists of spoken interactions in some form of language learning setting.

Based on the transformation that happens in the process of corpus building, we may either be concerned with the representation of multimodality extant in the source – e.g. whether a corpus of newspaper articles contains and/or makes searchable the images that were included in the articles – or multimodality in representation – e.g. whether a corpus of spoken interaction contains the medium which it represents (spoken language) in addition to the medium in which it primarily represents the population (written text). As a result of this duality, a multimodal corpus may refer to a corpus that makes available modes beyond writing directly, e.g. by including digitised audio and video files (see, e.g., Allwood 2008), or to text collections that have annotation layers encoding other modes in writing, e.g. by transcribing hand gestures or emojis (see, e.g., Collins 2020).

Finally, to complicate things further, modes themselves are only partially defined ontologically in terms of the semiotic mechanisms they use to make meaning, and are also shaped by social contexts and practices (Kress 2010: 86–87), e.g. when it comes to deciding whether the materiality of the book and the font used in printing are a meaningful part of the text and the language of a novel.

Given these complexities and the fact that ever more settings of language use are represented in ever more different ways in linguistic corpora, it is clear that we cannot discuss all the possibilities and challenges of multimodal corpora in this section. Instead, we will limit ourselves to a subjective take on three questions:

- (1) How are modes and media beyond writing included in corpora that are of interest to researchers in linguistic pragmatics?
- (2) What examples in terms of corpora and studies are there at the time of writing?
- (3) What challenges and opportunities lie ahead for multimodal corpus pragmatics?

## 7.2 Representing Modes and Media in Corpora

The composition of corpora always depends on the research they are intended for. Since pragmatics rests on the assumption that meaning depends on context, it is especially salient in the case of corpus pragmatic applications that decisions in the compilation of data collections for corpus research extend

not only to type and number of included texts but beyond that to the representation of all semiotic codes involved in the meaning-making of texts as situated language use.

Prototypical questions are, for instance, how film and sound images can be included in corpora of television language (Bednarek 2015; Section 5) or audiovisual translation (AVT) (Soffritti 2018); how images in newspapers are incorporated in news media corpora (Landert 2014; Bednarek & Caple 2017); how emojis are represented in corpora of Facebook communication (Collins 2020) or viewer comments (Messerli & Locher 2021; see Section 7.6); or how features of orality in spoken interaction are included in respective corpora (Adolphs & Carter 2013). The answers for the analyst will be specific not only to each of the genres mentioned in this exemplary list but also to the pragmatic question for which the corpus is intended. What they all share, however, is the underlying problem of modelling acts of communication as they occur in their natural habitat within the world of the corpus, and how to reach a level of representation that allows the pragmatic researcher to analyse aspects of illocution and perlocution based on the corpus representation of the communicative acts.

### 7.3 Application 1: Memes

As an illustration, we will point to the small *Swiss Memes Corpus (SMC)* that was compiled for the study published in Dynel and Messerli (2020), which was interested in a cross-cultural comparison of humorous depictions of Switzerland in Swiss, Polish and international memes. The data collected for the corpus consisted of only a few hundred memes, collected from three different sites, with each meme considered a multimodal text. As one of the simpler examples of multimodality, each of these texts incorporates an arrangement of some form of image and written text, which together create meaning. For the study, the *SMC* was encoded as a simple table, consisting of the image of the meme, its written text encoded as plain text, and manual coding of themes into which the researchers categorised each text. For the purposes of the study, which rested mainly on the qualitative analysis of typical examples, the composition of the *SMC* allowed a selection of examples based on theme, and access to the multimodal – albeit decontextualised – representation of the meme including its visual components.

If we were to be interested, however, in a more representative picture of how Switzerland is positioned in these and similar memes – perhaps based on a larger corpus of memes –, a corpus that is only searchable based on meme captions and themes might be insufficient. The issue is that *SMC*, while including the visual imagery of each meme, does not make it searchable. It may be

immediately recognisable, for instance, that many of the memes contain a Swiss flag, but that information is only usable for corpus analysis if the combination of encoding and corpus search interface allow access to it. For the different types of traditional corpora we have focused on up to this point in this Element, this means that we have to textually encode not only the captions but also aspects of the pictorial content of the memes.

## 7.4 Multimodality and Multimediality

If we stick to a traditional understanding of corpus linguistics, the main question of multimodality in corpora is thus not one of multimediality – how to include pictures or voice recordings together with written text – but of intersemiotic transfer: how can we encode more of the complex text and context of communicative acts in a way that is understandable by traditional corpus methods. Beyond traditional corpus designs, we may ask how we can expand corpus linguistic methods so they can systematically capture information that is not encoded in this way.

If multimodality is understood as a problem of encoding, multimodal corpora face similar problems to those of multimodal transcription (see, e.g., [Baldry & Thibault 2006](#); [Messerli 2020](#)), with the additional pressure that transcription from anything other than written language to writing needs to be done in a fashion that is understandable not only to human readers, but also defined clearly and consistently enough that it can be included as a well-structured annotation (e.g. in the form of XML tags or stand-off) that makes it machine-readable.

If multimodality is multimediality, then the burden lies on the development of more sophisticated corpus architectures that can make machine-readable even what has not been manually made so. A case in point is the VIAN-DH project currently in development at the University of Zurich, which combines image recognition (e.g. of hand gestures) with conversation analytic and corpus methods to create a new research tool as well as a multimedial corpus of multimodal communication in video clips. Potentially, this makes it possible for a researcher to identify a particular hand gesture in the video or even use their laptop camera to record the gesture they perform themselves and search the corpus for occurrences of visually similar gestures.

As the brief sample of available multimodal corpora in [Section 7.5](#) will show, such corpora are not yet widely available at the time of writing. The reason for this is the amount of expertise that is required for the compilation and maintenance of such corpora, but also the requirements for the infrastructure that hosts them. Instead, the multimodal corpora that exist employ modes beyond writing mostly to allow a more contextualised representation of language in use and/or

make select multimodal aspects available by means of manual coding (see, e.g., the description of the *NMMC* in [Adolphs & Carter 2013](#): 145–57).

## 7.5 Multimodal Corpora: A Sample of Currently Available Resources

To get an overview of available multimodal corpora, we can start with a cluster sample of those available on Europe’s *CLARIN* infrastructure. Given the scarcity of such corpora, we will not limit ourselves to English language corpora here. Out of seventeen multimodal corpora on *CLARIN*, fifteen are on different genres of spoken language, whereas only two, the *Multimodal Corpus of Tourist Brochures Produced by the City of Helsinki, Finland (1967–2008)* and the *Hindi Visual Genome 1.0*, contain multimodal representations of written texts. The former is a textbook case of the encoding route described in [Section 7.2](#) – apart from the brochures’ text, it contains annotation for various aspects, including layout, typography and rhetorical structure.

The spoken multimodal corpora on *CLARIN* are dedicated to a range of different speech events or genres – from friendly conversations among people that know each other well in the *IFA Dialog Video corpus* to simulated job interviews in the *Hungarian Multimodal Corpus*. In terms of annotation, the corpora follow different standards, but often include some form of annotation of gaze and/or gestures. All corpora are small in size and there are very few resources present for any given language. If we assume a researcher in English corpus pragmatics, it is worth noting that only two corpora contain samples of spoken English including some multimodal aspects – the *Eye-tracking in Multimodal Interaction Corpus* and the *Bielefeld Speech and Gesture Alignment Corpus (SaGA)*.

Beyond *CLARIN*, notable multimodal corpora include corpora that allow for the searching of speech phenomena, which includes the more specialised *Buckeye Corpus of Conversational Speech* and *Switchboard Corpus*, the *Scottish Corpus of Texts & Speech (SCOTS)*, which includes texts and many aligned audio recordings, and the more general *BNC Audio*, which not only provides aligned audio clips for about half the spoken section of the *BNC*, but also includes phonemic transcription and makes searchable aspects of phonology such as syllable counts and stress patterns (see [Hoffmann & Arndt-Lappe 2021](#)). Other, similar examples of transcriptions with aligned audio include the new *London–Lund Corpus 2* (British English) and the *Santa Barbara Corpus of Spoken American English*.

Within the field of language acquisition, *CHILDES* is a large collection of corpora which contain transcripts of child language acquisition as their main text and sometimes accompanying audio and/or video files for contextualisation and illustration. Within studies of interpreting, [Bernardini et al.](#)

(2018) provide an overview of the multimodal aspects of the *European Parliament Interpreting Corpus (EPIC)*, *EPICG* and *EPTIC* – three corpora of European Parliament plenary debates – and also incorporate a step-by-step guide of building a corpus that is not only multimodal, but also multilingual, thus requiring the alignment of both different modes and languages in order to allow the comparison of source and target languages in the context of interpreting (see Section 2.8).

Another important corpus pragmatic domain where multimodal corpora are central concerns the representation of sign languages. The volume edited by Fenlon and Hochgesang (2022) has recently mapped out in great detail the processes of building signed language corpora as well as their applications in existing and future research. Examples of multimodal sign language corpora mentioned are the early *Auslan Corpus*, with video recordings in the sign language Auslan, the *German Sign Language (DGS) Corpus* and the *British Sign Language (BSL) Corpus*. Interestingly, many of these corpora are not only available online but also provide detailed annotation guidelines, and in some cases templates for software like ELAN.

For the domain of text-image corpora, the *Lampeter Corpus of Early Modern English Tracts (LC)* is an example of a historical corpus that encodes visual aspects of the layout of historical tracts. In the context of computer-mediated communication, Christiansen et al. (2020) propose to employ what they term Visual Constituent Analysis (VCA) in the building of corpora containing, in particular, posts to social media. The importance of visual non-verbal components of online posts such as tweets cannot be overstated, and the simple example of searching tweets for names like ‘Clinton’ or ‘Trump’ shows that while maybe half the tweets about Donald Trump and Hillary Clinton in a given corpus can be found this way, the other half refers to either of the two US American politicians by means of a picture but without textual reference (Christiansen et al. 2020, working on the first iteration of the *Twitter Internet Research Agency Repository*). The way out proposed as part of VCA is to make use of commercially available automatic image annotation, e.g. Google Cloud Vision, to not only make texts in images readable by means of OCR, but also to enrich them with labels including named entities, recognised objects or parts of human bodies and faces, and even intertextual aspects that link individual posts to, for instance, other people and institutions. This is one way of encoding multimodality in monomodal written text, while avoiding the often prohibitive laboriousness of entirely manual coding (see Section 6).

While Christiansen et al.’s (2020) project again points to ways of addressing multimodality that take us beyond conventional corpus pragmatics, Section 7.6

illustrates the questions and challenges of multimodality based on a more traditional corpus-based study design.

### 7.6 Application 2: Humour Support in Viewer Comments

As part of a project dedicated to communicative practices in English fansubtitles and viewer comments to Korean TV drama, [Messerli and Locher \(2021\)](#) address what they have termed humour support indicators (HCIs) in a corpus of textual comments that fans write while streaming episodes on viki.com. The communicative setting modeled by the corpus consists of a viewer connecting to the streaming site through a web browser or a dedicated app on a phone, laptop or TV. Within the active window or app, there is video and audio, fansubtitles in the conventional position central at the bottom, and the comments themselves superimposed on the picture or in a separate window to its side. Finally, the comments themselves not only contain written text but also make abundant use of emojis.

The corpus design for the project consists of two corpora, one for the comments and one for the English fansubtitles. The two corpora are aligned based on the shared time code, relative to the episode streams, and the same time codes allow access to the entirety of semiotic modes on the streaming platform itself. The corpus of comments (*KTACC*) is designed as a monomodal collection, but contains unicode characters for the emojis, so that they, too, are searchable like words. For the purposes of the study, this makes it possible, for instance, to identify collocations of emojis and laugh particles – say ‘hahaha’ together with 😂 – or simple frequency searches to identify what emoji is employed by users most often. However, when the researchers wanted to identify icons that have a humorous meaning, manual identification of individual emojis was necessary, a step akin to identifying IFIDs for humour support.

Due to the frequency of emojis, this representation of the multimodality is as crucial for the understanding of humour support in viewer comments as the pictorial content in tweets highlighted by [Christiansen et al. \(2020\)](#). While this aspect was thus incorporated into *KTACC*, and the interplay between comments and subtitles was approached by means of a time-aligned corpus design, perhaps equally crucial relationships between the comments, subtitles and streamed videos are a blind spot for the corpus. Accordingly, most insights gained so far into these relationships have been due to exhaustive manual coding of samples ([Locher 2020; Locher & Messerli 2020](#)).

## 7.7 Conclusion

The corpus pragmatic community has, at the time of writing, only just begun to find ways of incorporating more of the multimodality that is crucial to any genre of situated language use into corpus designs. While corpora such as those outlined in [Section 7.5](#) may include multimediality, i.e. data beyond written language, and provide access to multimodality by means of (en)coding non-linguistic semiotic modes, for the time being corpora are strictly speaking almost universally monomodal representations of multimodal actions. This does in no way undermine the value of corpus pragmatic research – a model of reality that includes a reduction of complexity is arguably at the basis of all research – but it requires at least awareness that such a reduction has taken place, and ideally the exploration of new ways to incorporate other modes into new corpus designs, and other methods into corpus-based study designs.

The avenues we have outlined here constitute three main areas:

- (1) mixed-method studies that combine horizontal reading with corpus analyses;
- (2) studies that encode visual or audio information in plain text – typically manually by trained coders, but sometimes automatically, e.g. through machine-learning-based image recognition tools;
- (3) and potentially new corpus search interfaces that dynamically adapt to researchers' needs through processes like active learning, thus opening up new possibilities for the identification of communicative patterns in multimodal text.

Given recent developments in all three areas, we expect that multimodal corpus pragmatic research will gain a lot of importance in the next years and will likely lead to exciting new insights into language use in context.

## 8 Open Issues and Outlook

At the beginning of this Element, we argued that the advantages of doing corpus pragmatics can be found in pattern finding, systematicity, generalisation, reproducibility and transparency ([Section 1.6](#)), and throughout this Element we attempted to illustrate these key features. To conclude, the following subsections give a brief summary of the challenges and opportunities of corpus pragmatics that we discussed throughout the various sections of this Element. We include suggestions of how to engage with corpus pragmatics and we end with a brief outlook.

## 8.1 Corpus Methods for Function-to-Form Approaches

One of the areas in which we see great potential for future research is the development of corpus methods that are specifically tailored towards the study of pragmatic functions. Since the earliest days of corpus linguistics, researchers have used corpora to study pragmatic phenomena. Nevertheless, many of the tools that are most commonly used to search and analyse corpus data have been developed for the study of grammar and lexicon. This has led to an abundance of options to study forms in corpora: e.g. concordances, collocations, n-grams and keywords. All these options can be used in form-to-function approaches of corpus pragmatics. In contrast, the options for function-to-form approaches are much scarcer.

In this Element, we have pointed out several ways in which corpora can be used for function-to-form approaches. In [Section 4](#), we presented corpus-assisted approaches, in which researchers analyse small self-compiled datasets in a systematic manner and where corpus tools and qualitative data analysis software make it possible to quantify observations across different sections of the data. In [Section 3](#), we discussed data-driven approaches, where differences in pragmatic functions – such as persuasion – are identified through aspects that are an integral part of the original data – such as user annotations. And in [Section 6](#), we presented approaches in which well-known aspects of the typical distribution of pragmatic functions were used as a starting point for research: the placement of pragmatic functions in specific texts and parts of texts, and the tendency of pragmatic functions to cluster together. We think that exploring these and similar new methods that build closely on the typical characteristics of pragmatic functions holds a great deal of potential for the further development of corpus pragmatics.

## 8.2 Combining Quantitative and Qualitative Methods

Corpus approaches are predestined for quantification. Corpora include well-defined datasets that make it possible to identify linguistic features in a systematic manner. By comparing the frequency of such observations across different sections of data, we can draw conclusions about the distribution of such features in language use more generally. Thus, quantification is an integral part of corpus pragmatic methods. However, it would be problematic to reduce corpus pragmatics – or corpus linguistics in general – to quantification (see also [Egbert et al. 2020: ch. 7](#)). Qualitative analysis is equally crucial. In order to understand pragmatic functions, we need to engage in interpretations of language use in context, as we discussed in [Section 1](#). This means that both quantitative and qualitative steps are involved in corpus pragmatics.

Quantitative and qualitative methods can be combined in various ways. We may start with detailed manual analysis that focuses on the identification of patterns, which is then applied to the overall corpus (see [Section 4](#)). Alternatively, we can take as our starting point the overall frequency distribution of linguistic characteristics across different sets of data, and then study these characteristics in more detail. We discussed an example of such an approach in [Section 3](#). As a third option, [Section 6](#) discussed studies that use previous knowledge about pragmatics as a starting point to retrieve passages that can provide rich insight into the realisation of a pragmatic function when analysed qualitatively. The resulting observations can then be used for further quantitative and qualitative steps in the analysis.

### 8.3 Opportunities and Challenges of New Corpus Resources

Linguistic corpora have undergone a great deal of diversification since the early days of corpus linguistics. In [Section 2](#), we emphasised this point by illustrating some of the characteristics that distinguish corpora from each other. The characteristics we discuss there are not mutually exclusive, and they can combine in any number of ways to create unique new corpus resources. Some of these new kinds of corpora hold special potential for pragmatic research. For instance, pragmatically annotated corpora make it possible to search for pragmatic functions and to take into account factors that affect the realisation of pragmatic features. Another example is multimodal corpora, which we discussed in detail in [Section 7](#). By including sound, images and videos, such corpora allow researchers to study language use in its multimodal context. Finally, the rapid advances in computational linguistics create the potential for new ways of automatic and semi-automatic annotation, including annotation of pragmatic phenomena, and are likely to offer new approaches to the study of pragmatic phenomena in corpora.

As corpora become more diverse in structure and composition, new challenges for the compatibility and comparability of corpus resources will present themselves (see [Section 5](#)). Developing tools that support smooth and reliable working across different corpora without sacrificing metainformation and annotation will be crucial. Such tools should be open access in order to make it possible for researchers to control and adjust how corpus data is treated. Projects such as *AntConc*, *LancsBox* and *NoSketch Engine* are very valuable in this respect. As new kinds of corpus resources are being developed, it will be important to pay attention to the specific needs of pragmatic studies. These needs include, for instance, the option of accessing context beyond just a few words to the left and right of a search term. Equally crucial is the availability of

transparent metainformation on the origin of the data that is included in a corpus, as well as any factors that may influence how pragmatic features are realised.

#### 8.4 Data Awareness

Another issue we want to highlight is that corpus pragmaticists, like all corpus linguists in general, have a duty to inform themselves about the corpus they are working with (see Egbert et al. 2020: ch. 2). They need to know how the compilation of the corpus went about and what principles guided the sampling of texts. In Section 5, on comparability, we talked about this issue in relation to working with different corpora in combination. Even when working with just one corpus, it is equally important that users of a corpus know about what kind of data they are working with. If you are working with your own self-compiled data, you are in control and know what is ‘in’ the corpus. If you work with corpora provided by others, it is your duty as a scholar to inform yourself about its compilation so that you can better assess how this corpus is suitable for your research needs. Such information is usually accessible in publications about a corpus or online information that accompanies the corpus, although the amount of quality of information that is available varies a great deal. Especially for large corpora that integrate unmonitored data – often from online sources that are freely available – reliable information about the origin and context of the corpus data is often not provided to a sufficient extent.

For researchers compiling their own corpora, there are additional requirements with respect to the ethics of data collection. In Section 3.3, we have given some pointers to sources that help corpus pragmatics scholars in their attempt to make decisions concerning ethical conduct and heeding copyright concerns when compiling a corpus. Just like in any data collection, scholars in corpus pragmatics too need to be informed about ethics when using an existing corpus and take ethics considerations into account when compiling corpora from scratch. Especially for big corpora, incorporating ethics considerations remains a challenge and viable solutions have to be found for each corpus independently.

#### 8.5 Where to Start?

This Element has presented a brief summary of many different approaches. Consulting the references mentioned in the respective section is certainly a good starting point to begin a research project on any of them. For further practical examples and guidance, Rühlemann (2019) presents an introduction to different areas of corpus pragmatic research based on the *BNC*. For a more detailed

discussion of theoretical background, Aijmer and Rühlemann's (2014) handbook *Corpus Pragmatics* can be consulted, which provides a rich overview of relevant concepts and previous research.

As we emphasised throughout the Element, any corpus pragmatic study should start by engaging closely with the corpus, the data it includes and the principles according to which it was compiled. The list of corpora in [Appendix A](#) includes a large range of different corpora which have been used for pragmatic research. Any of these can be explored for additional aspects of pragmatics. [Appendix B](#) includes further tools that can be applied in corpus pragmatic studies, for instance when working with self-compiled corpora.

## 8.6 Creativity and Dynamics of the Field and Outlook

Corpus linguistics has been at the forefront of digital research in the humanities for more than forty years, and corpus pragmatics has enriched it through dynamic and creative contributions. We cannot stress enough that this field is dynamic in itself and calls for dynamic solutions at the same time. While we keep developing new ways of approaching pragmatic research challenges with existing corpora, new technological possibilities are being developed and are drawn on in order to tackle our research questions. These technical affordances are constantly evolving so that any new project design will face questions regarding whether to work with existing corpora (that might not be ideal in their composition or in the way one can work with them) or build new corpora (and engaging in a labour-intensive process of corpus creation and deciding how to ensure compatibility and comparability with existing resources). Establishing best practices and guidelines will remain crucial for as long as new corpora and tools are becoming available – which we are confident will be the case for the foreseeable future.

## Appendix A: Corpora

The following list presents all corpora mentioned in the Element, including information on where to access them and/or find additional information about them.

**ARCHER. A Representative Corpus of Historical English Registers.**

ARCHER Consortium. (1993–). [www.projects.alc.manchester.ac.uk/archer/](http://www.projects.alc.manchester.ac.uk/archer/).

**The Auslan Corpus.** Johnston, Trevor A. et al. (2008–). <https://auslan.org.au>.

**BASE. British Academic Spoken English Corpus.** Thompson, Paul & Nesi, Hilary. (2001). The British Academic Spoken English (BASE) Corpus Project. *Language Teaching Research* 5(3), 263–4. [www.sketchengine.eu/british-academic-spoken-english-corpus/](http://www.sketchengine.eu/british-academic-spoken-english-corpus/).

**BBC. Birmingham Blog Corpus.** (2010). Compiled by the Research and Development Unit for English Studies at Birmingham City University. [www.webcorp.org.uk/blogs](http://www.webcorp.org.uk/blogs).

**BNC1994. British National Corpus.** BNC Consortium. (1991–). [www.natcorp.ox.ac.uk/](http://www.natcorp.ox.ac.uk/).

**BNC2014. British National Corpus 2014.** <http://corpora.lancs.ac.uk/bnc2014/>.

**BNC Audio.** Audio part of the British National Corpus. See BNC1994.

**Brexit Corpus.** SENSEI-EU. University of Trento, Websays.com & Aix-Marseille University. (2016). [www.sense-eu.info/](http://www.sense-eu.info/), [www.sketchengine.eu/news/brexit-corpus-referendum/](http://www.sketchengine.eu/news/brexit-corpus-referendum/).

**Brown Corpus.** Kučera, Henry & Francis, W. Nelson. (1964/1979). <http://icame.uib.no/brown/bcm.html>.

**BSL. British Sign Language Corpus.** (2008–11). Cormier, Kearsy, Schembri, Adam, Fenlon, Jordan, et al. <https://bslcorpusproject.org>.

**Buckeye Corpus of Conversational Speech.** The Ohio State University. (2007, 2nd release). <https://buckeyecorpus.osu.edu/>.

**CANBEC. Cambridge and Nottingham Business English Corpus.** University of Nottingham, UK & Cambridge University Press. (2003). Not publicly available.

**CED. Corpus of English Dialogues 1560–1760.** Kytö, Merja & Culpeper, Jonathan. (2006). Oxford Text Archive, <http://hdl.handle.net/20.500.12024/2507>.

**CEEM. Corpus of Early English Medical Writing.** Taavitsainen, Irma, Pahta, Päivi & Mäkinen, Martti. (1995–2019).

MEMT. Middle English Medical Texts 1375 – 1500. (2005). John Benjamins.

- EMEMT. Early Modern English Medical Texts 1500–1700. (2010). John Benjamins.
- LMEMT. Late Modern English Medical Texts 1700–1800. (2019). John Benjamins.
- CHELAR. Corpus of Historical English Law Reports 1535–1999.** Rodríguez-Puente, Paula, Fanego, Teresa, José López-Couso, María, Méndez-Naya, Belén & Núñez-Pertejo, Paloma. (2016). [www.usc-vlcg.es/CHELAR.htm](http://www.usc-vlcg.es/CHELAR.htm).
- CHILDES. Child Language Data Exchange System.** MacWhinney, Brian. (1984). <https://childe.talkbank.org/>.
- CLMET. The Corpus of Late Modern English Texts, version 3.0.** Diller, Hans-Jürgen, De Smet, Hendrik & Tyrkkö, Jukka. (2011). [https://perswww.kuleuven.be/~u0044428/clmet3\\_0.htm](https://perswww.kuleuven.be/~u0044428/clmet3_0.htm).
- COCA. The Corpus of Contemporary American English.** Davies, Mark. (2008–). [www.english-corpora.org/coca/](http://www.english-corpora.org/coca/).
- COHA. The Corpus of Historical American English.** Davies, M. (2010). [www.english-corpora.org/coha/](http://www.english-corpora.org/coha/).
- Corpus of Czech Students' Spoken English.** Šárka Ježková, Univerzita Pardubice. (2015). <https://bit.ly/3iPw9fT>.
- DGS. Corpus of German Sign Language.** (2009–23). Hanke, Thomas, Herrmann, Annika, Rathmann, Christian et al. [www.sign-lang.uni-hamburg.de/](http://www.sign-lang.uni-hamburg.de/).
- DOE. Dictionary of Old English Corpus.** (1981/2009). [www.doe.utoronto.ca](http://www.doe.utoronto.ca).
- EEBO. Early English Books Online.** <https://proquest.libguides.com/eebopqp>.
- EMEMT.** See CEEM.
- EPIC. European Parliament Interpreting Corpus.** (2011). <https://cris.unibo.it/handle/11585/132580>.
- EUROPARL. European Parliament Proceedings Parallel Corpus 1996–2011.** [www.statmt.org/europarl/](http://www.statmt.org/europarl/).
- Eye-tracking in Multimodal Interaction Corpus.** Holler, Judith & Kendrick, Kobin. (2013–14). MPI for Psycholinguistics Archive. <https://bit.ly/3VLAuzi>.
- GLBCC. Giessen–Long Beach Chaplin Corpus.** Jucker, Andreas H., Müller, Simone & Smith, Sara. (2006). Oxford Text Archive. <http://hdl.handle.net/20.500.12024/2506>.
- GloWbE. Corpus of Global Web-Based English.** Davies, Mark. (2013). [www.english-corpora.org/glowbe/](http://www.english-corpora.org/glowbe/).
- HC. Helsinki Corpus of English Texts.** (1991). Oxford Text Archive, <http://hdl.handle.net/20.500.12024/1477>. (2011). <https://helsinki-corpus.arts.gla.ac.uk/display.py?what=index>.
- Hindi Visual Genome 1.0.** Parida, Shantipriya, Bojar, Ondrej & Dash, Satya Ranjan. (2019). <https://ufal.mff.cuni.cz/hindi-visual-genome>.

- Hungarian Multimodal Corpus.** (2013). <https://hdl.handle.net/1839/00-0000-0000-001A-E17C-1>.
- ICE. International Corpus of English.** (1990–). [www.ice-corpora.uzh.ch/en.html](http://www.ice-corpora.uzh.ch/en.html).
- ICLE. International Corpus of Learner English.** Granger, Sylviane. (2002/2009). <https://uclouvain.be/en/research-institutes/ilc/cecl/icle.html>.
- IFA Dialog Video corpus.** Nederlandse Taal Unie. (2007). [www.fon.hum.uva.nl/IFA-SpokenLanguageCorpora/IFADVcorpus/](http://www.fon.hum.uva.nl/IFA-SpokenLanguageCorpora/IFADVcorpus/).
- KTACC. Korean television drama Time-Aligned Comments Corpus.** Locher, Miriam & Messerli, Thomas. Not publicly available.
- LC. The Lampeter Corpus of Early Modern English Tracts.** Schmid, Josef, Claridge, Claudia & Siemund, Rainer. (1999). <http://korpus.uib.no/icame/manuals/LAMPETER/LAMPHOME.HTM>.
- LINDSEI. Louvain International Database of Spoken English Interlanguage.** Gilquin, Gaëtanelle, De Cock, Sylvie & Granger, Sylviane. (2010). <https://bit.ly/3Bmhr6t>.
- LLC. London-Lund Corpus of Spoken English.** (1990). Svartvik, Jan. Oxford Text Archive. [http://hdl.handle.net/20.500.12024/0168](https://hdl.handle.net/20.500.12024/0168).
- LLC2. London–Lund Corpus 2.** (in progress). <https://projekt.ht.lu.se/llc2>.
- LMEMT.** See CEEM.
- LOB. Lancaster–Oslo/Bergen Corpus.** (1978). Leech, Geoffrey, Johansson, Stig & Hofland, Knut. <http://korpus.uib.no/icame/manuals/LOB/INDEX.HTM>.
- Longman Corpus of Spoken and Written English.** [www.pearsonlongman.com/dictionaries/corpus/index.html](http://www.pearsonlongman.com/dictionaries/corpus/index.html).
- MEMT.** See CEEM.
- MICASE. The Michigan Corpus of Academic Spoken English.** Simpson, R. C., Briggs, S. L., Ovens, J. & Swales, J. M. (2002). Ann Arbor, MI: The Regents of the University of Michigan. <https://quod.lib.umich.edu/m/micase/>.
- Movie Corpus.** Davies, Mark. (2019). [www.english-corpora.org/movies/](http://www.english-corpora.org/movies/).
- Multimodal Corpus of Tourist Brochures Produced by the City of Helsinki, Finland (1967–2008).** Hippala, Tuomo. (2015). <http://urn.fi/urn:nbn:fi:lb-201411281>.
- NMMC. Nottingham Multimodal Corpus.** See Adolphs & Carter (2013) for further information.
- NOW. The Corpus of News on the Web.** Davies, Mark. (2016–). [www.english-corpora.org/now/](http://www.english-corpora.org/now/).
- OBC. Old Bailey Corpus 2.0.** Huber, Magnus, Nissel, Magnus & Puga, Karin. (2016). hdl:11858/00-246 C-0000-0023-8CFB-2 / [www1.uni-giessen.de/oldbaileycorpus/](http://www1.uni-giessen.de/oldbaileycorpus/). [www.oldbaileyonline.org/static/Project.jsp](http://www.oldbaileyonline.org/static/Project.jsp).

- PCEEC. Parsed Corpus of Early English Correspondence.** CEEC Project Team. (2006). Oxford Text Archive. <http://hdl.handle.net/20.500.12024/2510>.
- SaGA. Bielefeld Speech and Gesture Alignment Corpus.** [www.phonetik.uni-muenchen.de/Bas/BasSaGAeng.html](http://www.phonetik.uni-muenchen.de/Bas/BasSaGAeng.html).
- SBC. Santa Barbara Corpus of Spoken American English.** Du Bois, John W., Chafe, Wallace L., Meyer, Charles, et al. (2000–2005). [www.linguistics.ucsb.edu/research/santa-barbara-corpus](http://www.linguistics.ucsb.edu/research/santa-barbara-corpus).
- SCOTS. Scottish Corpus of Texts & Speech.** Kay, Christian, Thompson, Henry, Corbett, John, et al. (2002–4). [www.scottishcorpus.ac.uk/](http://www.scottishcorpus.ac.uk/).
- SMC. Swiss Memes Corpus.** Dynel, Marta & Messerli, Thomas. Not publicly available.
- SOAP. Corpus of American Soap Operas.** Davies, Mark. (2011–). [www.english-corpora.org/soap/](http://www.english-corpora.org/soap/).
- SPC. Sociopragmatic Corpus; part of CED. Corpus of English Dialogues 1500–1700.** Not publicly available.
- SPICE-Ireland Corpus.** Kirk, John M. & Kallen, Jeffrey L. (2011). <https://johnmkirk.etinu.net/cgi-bin/generic?instanceID=11>.
- Switchboard Corpus.** Godfrey, John J. & Holliman, Edward. (1992–7). <https://catalog.ldc.upenn.edu/LDC97S62>.
- SydTV. Sydney Corpus of Television Dialogue.** Bednarek, Monika. (2020). <https://cqpw-prod.vip.sydney.edu.au/CQPweb/>.
- TV Corpus.** Davies, Mark. (2019). [www.english-corpora.org/tv/](http://www.english-corpora.org/tv/).
- Twitter Internet Research Agency Repository (First Iteration).** (2018). <https://archive.org/details/twitter-ira>.
- VOICE. Vienna-Oxford International Corpus of English 3.0 online.** (2021). <https://voice.acdh.oeaw.ac.at/>.

## Appendix B: Corpus Tools and Additional Resources

The following list provides an overview of corpus linguistic tools and platforms that were mentioned throughout the Element.

**AntConc.** [www.laurenceanthony.net/software/antconc/](http://www.laurenceanthony.net/software/antconc/).

**ATLAS.ti.** <https://atlasti.com/>.

**BNClab.** <http://corpora.lancs.ac.uk/bnclab/search>.

**CATMA.** Computer Assisted Text Markup and Analysis. <https://catma.de/>.

**CLARIN.** Common Language Resources and Technology Infrastructure.  
[www.clarin.eu/](http://www.clarin.eu/).

**CoRD.** Corpus Resource Database. <https://varieng.helsinki.fi/CoRD/index.html>.

**CQPweb.** Web-based graphical user interface for CWB. <https://cwb.sourceforge.io/cqpweb.php>.

**CWB.** IMS Open Corpus Workbench. <https://cwb.sourceforge.io/>.

**DART.** Dialogue Annotation and Research Tool. See Weisser (2014).

**Google Ngram Viewer.** <https://books.google.com/ngrams>.

**koRpus.** Text analysis package for R. <https://cran.r-project.org/web/packages/koRpus/index.html>.

**LancsBox.** Lancaster University Corpus Toolbox. <http://corpora.lancs.ac.uk/lancsbox/>.

**MAXQDA.** [www.maxqda.de/](http://www.maxqda.de/).

**NLTK.** Natural Language Toolkit. [www.nltk.org/](http://www.nltk.org/).

**NoSketch Engine.** <https://nlp.fi.muni.cz/trac/noske>.

**NVivo.** <https://bit.ly/3HueNzk>.

**OTA.** Oxford Text Archive. <https://ota.bodleian.ox.ac.uk/repository/xmlui/>.

**Oxygen XML Editor.** [www.oxygenxml.com/](http://www.oxygenxml.com/).

**polmineR.** R-based package for corpus analysis using the CWB. <https://cran.r-project.org/web/packages/polmineR/index.html>.

**quanteda.** Quantitative Analysis of Textual Data. <https://quanteda.io/>.

**Sketch Engine.** [www.sketchengine.eu/](http://www.sketchengine.eu/).

**spaCy.** Python library for natural language processing. <https://spacy.io/>.

**SPSS.** IBM SPSS Statistics Platform. [www.ibm.com/ch-de/products/spss-statistics](http://www.ibm.com/ch-de/products/spss-statistics).

**TXM.** Open source text analysis software. <https://txm.gitpages.huma-num.fr/textometrie/en/index.html>.

**WordSmith Tools.** [www.lexically.net/wordsmith/](http://www.lexically.net/wordsmith/).

## References

- Adolphs, Svenja. (2008). *Corpus and Context: Investigating Pragmatic Functions in Spoken Discourse*. Amsterdam: John Benjamins.
- Adolphs, Svenja & Carter, Ronald. (2013). *Spoken Corpus Linguistics: From Monomodal to Multimodal*. New York: Routledge.
- Aijmer, Karin. (1997). *I think – An English modal particle*. In Swan, Toril & Westvik, Olaf Jansen, eds., *Modality in Germanic Languages: Historical and Comparative Perspectives*. Berlin: Mouton de Gruyter, pp. 1–47.
- Aijmer, Karin. (2002). *English Discourse Particles: Evidence from a Corpus*. Amsterdam: John Benjamins.
- Aijmer, Karin. (2008). At the interface between grammar and discourse: A corpus-based study of some pragmatic markers. In Romero-Trillo, Jesús, ed., *Pragmatics and Corpus Linguistics: A Mutualistic Entente*. Berlin: De Gruyter Mouton, pp. 11–36.
- Aijmer, Karin. (2013). *Understanding Pragmatic Markers: A Variational Pragmatic Approach*. Edinburgh: Edinburgh University Press.
- Aijmer, Karin & Rühlemann, Christoph, eds. (2014). *Corpus Pragmatics: A Handbook*. Cambridge: Cambridge University Press.
- Allwood, Jens. (2008). Multimodal Corpora. In Lüdeling, Anke & Kytö, Merja, eds., *Corpus Linguistics: An International Handbook* (Vol. 1). Berlin: De Gruyter Mouton, pp. 207–25.
- Andersen, Gisle. (2011). Corpus-based pragmatics I: Qualitative studies. In Bublitz, Wolfram & Norrick, Neal R., eds., *Foundations of Pragmatics* (Handbook of Pragmatics Series 1). Berlin: Walter de Gruyter, pp. 587–627.
- Archer, Dawn. (2005). *Questions and Answers in the English Courtroom (1640–1760)* (Pragmatics & Beyond New Series 135). Amsterdam: John Benjamins.
- Archer, Dawn & Gillings, Matthew. (2020). Depictions of deception: A corpus-based analysis of five Shakespearean characters. *Language and Literature* 29(3), 1–29. <https://doi.org/10.1177/0963947020949439>.
- Baker, Paul. (2006). *Using Corpora in Discourse Analysis* (Continuum Discourse Series). London: Continuum.
- Baldry, Anthony, & Thibault, Paul J. (2006). *Multimodal Transcription and Text Analysis*. London: Equinox.
- Bayley, Paul & Williams, Geoffrey, eds. (2012). *European Identity: What the Media Say*. Oxford: Oxford University Press.

- Bednarek, Monika. (2015). Corpus-assisted multimodal discourse analysis of television and film narratives. In Baker, Paul & McEnery, Tony, eds., *Corpora and Discourse Studies*. Basingstoke: Palgrave Macmillan, pp. 63–87.
- Bednarek, Monika. (2018). *Language and Television Series: A Linguistic Approach to TV Dialogue* (Cambridge Applied Linguistics). Cambridge: Cambridge University Press.
- Bednarek, Monika, & Caple, Helen. (2017). *The Discourse of News Values*. Oxford: Oxford University Press.
- Bednarek, Monika, Veirano Pinto, Marcia & Werner, Valentin. (2021). Corpus approaches to telecinematic language: Introduction. *International Journal of Corpus Linguistics* 26(1), 1–9.
- Beeching, Kate. (2016). *Pragmatic Markers in British English: Meaning in Social Interaction*. Cambridge: Cambridge University Press.
- Bernardini, Silvia, Ferraresi, Adriano, Russo, Mariachiara, Collard, Camille & Defrancq, Bart. (2018). Building interpreting and intermodal corpora: A how-to for a formidable task. In Russo, Mariachiara, Bendazzoli, Claudio & Defrancq, Bart, eds., *Making Way in Corpus-Based Interpreting Studies*. Singapore: Springer, pp. 21–42.
- Biber, Douglas. (1988). *Variation across Speech and Writing*. Cambridge: Cambridge University Press.
- Biber, Douglas. (1993). Representativeness in corpus design. *Literary and Linguistic Computing* 8(4), 243–257.
- Blaette, Andreas. (2020). *polmineR: Verbs and Nouns for Corpus Analysis*. <https://doi.org/10.5281/zenodo.4042093>, R package version 0.8.2.
- Brinton, Laurel J. (2012). Historical pragmatics and corpus linguistics: Problems and strategies. In Kytö, Merja, ed., *English Corpus Linguistics: Crossing Paths*. Amsterdam: Rodopi, pp. 101–31.
- Burr, Vivien. (1995). *An Introduction to Social Constructionism*. London: Routledge.
- Butler, Christopher S. (2008). The subjectivity of *basically* in British English: A corpus-based study. In Romero-Trillo, Jesús, ed., *Pragmatics and Corpus Linguistics: A Mutualistic Entente*. Berlin: Mouton de Gruyter, pp. 37–63.
- Buyssse, Lieven. (2012). *So* as a multifunctional discourse marker in native and learner speech. *Journal of Pragmatics* 44(13), 1764–82.
- Buyssse, Lieven. (2017). The pragmatic marker *you know* in learner English. *Journal of Pragmatics* 121, 40–57.
- Buyssse, Lieven. (2020). “It was a bit stressy as well actually”: The pragmatic markers *actually* and *in fact* in spoken learner English. *Journal of Pragmatics* 156, 28–40.

- Cartoni, Bruno, Zufferey, Sandrine & Meyer, Thomas. (2013). Using the Europarl corpus for cross-linguistic research. *Belgian Journal of Linguistics* 27(1), 23–42.
- Christiansen, Alex, Dance, William & Wild, Alexander. (2020). Constructing corpora from images and text: An introduction to Visual Constituent Analysis. In Rüdiger, Sofia & Dayter, Daria, eds., *Corpus Approaches to Social Media*. Amsterdam: John Benjamins, pp. 149–74.
- Collins, Luke. (2020). Working with images and emoji in the Dukki Facebook Corpus. In Rüdiger, Sofia & Dayter, Daria, eds., *Corpus Approaches to Social Media*. Amsterdam: John Benjamins, pp. 175–96.
- Culpeper, Jonathan & Archer, Dawn. (2008). Requests and directness in Early Modern English trial proceedings and play texts, 1640–1760. In Jucker, Andreas H. & Taavitsainen, Irma, eds., *Speech Acts in the History of English* (Pragmatics & Beyond New Series 176). Amsterdam: John Benjamins, pp. 45–84.
- Culpeper, Jonathan & Kytö, Merja. (2010). *Early Modern English Dialogues: Spoken Interaction as Writing*. Cambridge: Cambridge University Press.
- Dayter, Daria. (2016). *Discursive Self in Microblogging: Speech Acts, Stories and Self-Praise*. Amsterdam: John Benjamins.
- Dayter, Daria. (2018). Self-praise online and offline: The hallmark speech act of social media? *Internet Pragmatics* 1(1), 184–203.
- Dayter, Daria. (2021). Dealing with interactionally risky speech acts in simultaneous interpreting: The case of self-praise. *Journal of Pragmatics* 174, 28–42.
- Dayter, Daria, & Messerli, Thomas C. (2022). Persuasive language and features of formality on the r/ChangeMyView subreddit. *Internet Pragmatics* 5(1), 165–95. <https://doi.org/10.1075/ip.00072.day>.
- Dayter, Daria & Rüdiger, Sofia. (2022). *The Language of Pick-Up Artists: Online Discourses of the Seduction Industry*. London: Routledge.
- DeCapua, Andrea, & Dunham, Joan Findlay. (2012). ‘It wouldn’t hurt if you had your child evaluated’: Advice to mothers in responses to vignettes from a US teaching context. In Limberg, Holger & Locher, Miriam A., eds., *Advice in Discourse*. Amsterdam: John Benjamins, pp. 73–96.
- Deutschmann, Mats. (2003). *Apologising in British English* (Skrifter Från Moderna Språk 10). Umeå: Institutionen för moderna språk, Umeå University.
- Doval, Irene & Sánchez Nieto, Teresa. (2019). *Parallel Corpora for Contrastive and Translation Studies*. Amsterdam: John Benjamins.
- Dynel, Marta, & Messerli, Thomas C. (2020). On a cross-cultural memescape: Switzerland through nation memes from within and from the outside. *Contrastive Pragmatics* 1(2), 1–32.

- Egbert, Jesse, Larsson, Tove & Biber, Douglas. (2020). *Doing Linguistics with a Corpus. Methodological Considerations for the Everyday User* (Elements in Corpus Linguistics). Cambridge: Cambridge University Press.
- Evert, Stefan & Hardie, Andrew. (2011). Twenty-first century corpus workbench: Updating a query architecture for the new millennium. *Proceedings of the Corpus Linguistics 2011 Conference*, University of Birmingham, UK.
- Fenlon, Jordan, & Hochgesang, Julie A. (eds.) (2022). *Signed Language Corpora*. Washington, DC: Gallaudet University Press.
- Franzke, Aline, Bechmann, Anja, Zimmer, Michael, Ess, Charles & the Association of Internet Researchers. (2020). Internet Research: Ethical Guidelines 3.0. <https://aoir.org/reports/ethics3.pdf>.
- Garcés-Conejos Blitvich, Pilar & Sifianou, Maria. (2019). Im/politeness and discursive pragmatics. *Journal of Pragmatics* 145, 91–101.
- Gast, Volker. (2015). On the use of translation corpora in contrastive linguistics. *Languages in Contrast* 15(1), 4–33.
- Gilquin, Gaëtanelle. (2008). Hesitation markers among EFL learners: Pragmatic deficiency or difference? In Romero-Trillo, Jesús, ed., *Pragmatics and Corpus Linguistics: A Mutualistic Entente*. Berlin: Mouton de Gruyter, pp. 119–49.
- Gray, Bethany, Biber, Douglas & Hiltunen, Turo. (2011). The expression of stance in early (1665–1712) publications of the *Philosophical Transactions* and other contemporary medical prose: Innovations in a pioneering discourse. In Taavitsainen, Irma & Pahta, Päivi, eds., *Medical Writing in Early Modern English* (Studies in English Language). Cambridge: Cambridge University Press, pp. 221–57.
- Guest, Greg, MacQueen, Kathleen M. & Namey, Emily E. (2012). *Applied Thematic Analysis*. Los Angeles, CA: SAGE Publications.
- Harrison, Simon, Todd, Zazie & Lawton, Rebecca. (2008). Talk about terrorism and the media: Communicating with the conduit metaphor. *Communication, Culture and Critique* 1(4), 378–95.
- Haugh, Michael. (2018). Corpus-based metapragmatics. In Jucker, Andreas H., Schneider, Klaus P. & Bublitz, Wolfram, eds., *Methods in Pragmatics* (Handbooks of Pragmatics 10). Berlin: De Gruyter Mouton, pp. 587–618.
- Heinrich, Philipp & Schäfer, Fabian. (2018). Extending corpus-based discourse analysis for exploring Japanese social media. In Tono, Yukio & Isahara, Hitoshi, eds., *Proceedings of 4th Asia Pacific Corpus Linguistics Conference (APCLC2018)*, pp. 135–40.
- Hilbert, Michaela & Krug, Manfred. (2010). The compilation of ICE Malta: State of the art and challenges along the way. *ICAME Journal* 34, 54–63.

- Hoffmann, Sebastian, & Arndt-Lappe, Sabine. (2021). Better data for more researchers: Using the audio features of BNCweb. *ICAME Journal* 45(1), 125–54. <https://doi.org/10.2478/icame-2021-0004>.
- Huschová, Petra. (2021). Modalized speech acts in a spoken learner corpus: The case of *can* and *could*. *Topics in Linguistics*, 22(1), 27–37. <https://doi.org/10.2478/topling-2021-0003>.
- Ifukor, Presley. (2010). “Elections” or “selections”? Blogging and twitting the Nigerian 2007 general elections. *Bulletin of Science, Technology & Society* 30(6), 398–414.
- Jacobs, Andreas & Jucker, Andreas H. (1995). The historical perspective in pragmatics. In Jucker, Andreas H., ed., *Historical Pragmatics. Pragmatic Developments in the History of English* (Pragmatics and Beyond New Series 35). Amsterdam: John Benjamins, pp. 3–33.
- Jucker, Andreas H. (2004). Gutenberg und das Internet. Der Einfluss von Informationsmedien auf Sprache und Sprachwissenschaft. *Networx* 40. [www.mediensprache.net/de/networx/docs/networx-40.aspx](http://www.mediensprache.net/de/networx/docs/networx-40.aspx).
- Jucker, Andreas H. (2013). Corpus Pragmatics. In Östman, Jan-Ola & Verschueren, Jef, eds., *Handbook of Pragmatics*. Amsterdam: John Benjamins, pp. 1–17.
- Jucker, Andreas H. (2018). Apologies in the history of English: Evidence from the Corpus of Historical American English (COHA). *Corpus Pragmatics* 2 (4), 375–98.
- Jucker, Andreas H. (2020). *Politeness in the History of English. From the Middle Ages to the present day*. Cambridge: Cambridge University Press.
- Jucker, Andreas H. & Kopaczek, Joanna. (2017). Historical (Im)politeness. In Culpeper, Jonathan, Haugh, Michael & Kádár, Dániel Z., eds., *The Palgrave Handbook of Linguistic (Im)politeness*. London: Palgrave Macmillan, pp. 433–59.
- Jucker, Andreas H. & Taavitsainen, Irma. (2008). Apologies in the history of English. Routinized and lexicalized expressions of responsibility and regret. In Jucker, Andreas H. & Taavitsainen, Irma, eds., *Speech Acts in the History of English* (Pragmatics and Beyond New Series 176). Amsterdam: John Benjamins, pp. 229–44.
- Jucker, Andreas H. & Taavitsainen, Irma. (2014). Complimenting in the history of American English: A metacommunicative expression analysis. In Taavitsainen, Irma, Jucker, Andreas H. & Tuominen, Jukka, eds., *Diachronic Corpus Pragmatics* (Pragmatics & Beyond New Series 243). Amsterdam: John Benjamins, pp. 257–76.
- Jucker, Andreas H., Schneider, Gerold, Taavitsainen, Irma & Breustedt, Barb. (2008). Fishing for compliments: Precision and recall in corpus-linguistic

- compliment research. In Jucker, Andreas H. & Taavitsainen, Irma, eds., *Speech Acts in the History of English* (Pragmatics & Beyond New Series 176). Amsterdam: John Benjamins, 273–94.
- Jucker, Andreas H., Schneider, Klaus P. & Bublitz, Wolfram, eds. (2018). *Methods in Pragmatics* (Handbooks of Pragmatics 10). Berlin: De Gruyter Mouton.
- Jucker, Andreas H., Schreier, Daniel & Hundt, Marianne, eds. (2009). *Corpora: Pragmatics and Discourse. Papers from the 29th International Conference on English Language Research on Computerized Corpora (ICAME 29)*. Ascona, Switzerland, 14–18 May 2008. Amsterdam: Rodopi.
- Kaindl, Klaus. (2013). Multimodality and translation. In Millán, Carmen & Bartrina, Francesca, eds., *The Routledge Handbook of Translation Studies*. London: Routledge, pp. 257–69.
- Kärkkäinen, Elise. (2003). *Epistemic Stance in English Conversation. A Description of its Interactional Functions, with a Focus on I think* (Pragmatics & Beyond New Series 115). Amsterdam: John Benjamins.
- Kim, Kyung H. (2014). Examining US news media discourses about North Korea: A corpus-based critical discourse analysis. *Discourse & Society* 25 (2), 221–44.
- Kirk, John M. (2015). *Kind of and sort of*: Pragmatic discourse markers in the SPICE-Ireland Corpus. In Amador-Moreno, Carolina P., McCafferty, Kevin & Vaughan, Elaine, eds., *Pragmatic Markers in Irish English*, 89–113. Amsterdam: John Benjamins.
- Koehn, Philipp. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*. Phuket, Thailand, pp. 79–86. [https://aclanthology.org/volumes/2005\\_mtsummit-papers/](https://aclanthology.org/volumes/2005_mtsummit-papers/).
- Koene, Ansgar & Adolphs, Svenja. (2015). Ethics considerations for corpus linguistic studies using internet resources. Working paper, HORIZON Digital Economy Research. [http://casma.wp.horizon.ac.uk/wp-content/uploads/2015/04/CL2015-CorpusLinguisticsEthics\\_KoeneAdolphs.pdf](http://casma.wp.horizon.ac.uk/wp-content/uploads/2015/04/CL2015-CorpusLinguisticsEthics_KoeneAdolphs.pdf).
- Kohnen, Thomas. (1997). Toward a theoretical foundation of “text type” in diachronic corpora: Investigations with the Helsinki Corpus. In Hickey, Raymond & Kytö, Merja, eds., *Tracing the Trail of Time. Proceedings from the Second Diachronic Corpora Workshop. New College, University of Toronto, Toronto, May 1995*. Amsterdam: Rodopi, pp. 185–97.
- Kohnen, Thomas. (2000). Corpora and speech acts: The study of performatives. In Mair, Christian, & Hundt, Marianne, eds., *Corpus Linguistics and Linguistic Theory. Papers from the Twentieth International Conference on English Language Research on Computerized Corpora (ICAME 20)*. Amsterdam: Rodopi, pp. 177–86.

- Koteyko, Nelya, Nerlich, Brigitte, Crawford, Paul & Wright, Nick. (2008). “Not rocket science” or “no silver bullet”? Media and government discourses about MRSA and cleanliness. *Applied Linguistics* 29(2), 223–43.
- Krendel, Alexandra, McGlashan, Mark & Koller, Veronika. (2022). The representation of gendered social actors across five manosphere communities on Reddit. *Corpora* 17(2), 291–321.
- Kress, Gunther R. (2010). *Multimodality. A Social Semiotic Approach to Contemporary Communication*. London: Routledge.
- Kress, Gunther R. & Van Leeuwen, Theo. (2001). *Multimodal Discourse: The Modes and Media of Contemporary Communication*. London: Arnold.
- Krishnamurthy, Ramesh. (1996). Ethnic, racial and tribal: The language of racism? In Caldas-Coulthard, Carmen & Coulthard, Malcolm, eds., *Texts and Practices: Readings in Critical Discourse Analysis*. London: Routledge, pp. 129–49.
- Landert, Daniela. (2014). *Personalisation in Mass Media Communication: British Online News between Public and Private* (Pragmatics & Beyond New Series 240). Amsterdam: John Benjamins.
- Landert, Daniela. (2019). Function-to-form mapping in corpora: Historical corpus pragmatics and the study of stance expressions. In Suhr, Carla, Nevalainen, Terttu & Taavitsainen, Irma, eds., *From Data to Evidence in English Language Research* (Language and Computers 83). Leiden: Brill, pp. 169–90.
- Leech, Geoffrey & Weisser, Martin. (2003). Generic speech act annotation for task-oriented dialogue. In Archer, Dawn, Rayson, Paul, Wilson, Andrew & McEnery, Tony, eds., *Proceedings of the Corpus Linguistics 2003 Conference* (Vol. 16). Lancaster: UCREL Technical Papers, pp. 441–6.
- Locher, Miriam A. (2006). *Advice Online. Advice-Giving in an American Internet Health Column*. Amsterdam: John Benjamins.
- Locher, Miriam A. (2020). Moments of relational work in English fan translations of Korean TV drama. *Journal of Pragmatics* 170, 139–55. <https://doi.org/10.1016/j.pragma.2020.08.002>.
- Locher, Miriam A., & Graham, Sage L. (2010). Introduction to Interpersonal Pragmatics. In Locher, Miriam A. & Graham, Sage L., eds., *Interpersonal Pragmatics*. Berlin: Mouton, pp. 1–13.
- Locher, Miriam A., & Messerli, Thomas C. (2020). Translating the other: Communal TV watching of Korean TV drama. *Journal of Pragmatics*, 170, 20–36. <https://doi.org/10.1016/j.pragma.2020.07.002>.
- Locher, Miriam A., & Schnurr, Stephanie. (2017). (Im)politeness in health settings. In Culpeper, Jonathan, Haugh, Michael & Kádár, Dániel, eds., *Palgrave Handbook of Linguistic (Im)Politeness*. London: Palgrave, pp. 689–711.

- Locher, Miriam A., & Thurnherr, Franziska. (2017). Typing yourself healthy: Introduction to the special issue on language and health online. *Linguistics Online*, 87(8/17), 3–24. <http://dx.doi.org/10.13092/lo.87.4170>.
- Love, Robbie & Baker, Paul. (2015). The hate that dare not speak its name? *Journal of Language Aggression and Conflict*, 3(1), 57–86.
- Lutzky, Ursula. (2012). *Discourse Markers in Early Modern English* (Pragmatics & Beyond New Series 227). Amsterdam: John Benjamins.
- Lutzky, Ursula & Kehoe, Andrew. (2017a). “I apologise for my poor blogging”: Searching for apologies in the Birmingham Blog Corpus. *Corpus Pragmatics* 1(1), 37–56.
- Lutzky, Ursula & Kehoe, Andrew. (2017b). “Oops, I didn’t mean to be so flippant”: A corpus pragmatic analysis of apologies in blog data. *Journal of Pragmatics* 116, 27–36.
- Lutzky, Ursula & Lawson, Robert. (2019). Gender politics and discourses of #mansplaining, #manspreading, and #manterruption on Twitter. *Social Media + Society* 5(3), 1–12. <https://doi.org/10.1177/2056305119861807>.
- MacQueen, Kathleen M., Mclellan-Lemal, Eleanor, Bartholow, Kelly & Milstein, Bobby. (2008). Team-based codebook development: Structure, process, and agreement. In Guest, Greg & MacQueen, Kathleen M., eds., *Handbook for Team-Based Qualitative Research*. Lanham: ALTAMIRA, pp. 119–36.
- Marchi, Anna & Taylor, Charlotte. (2009). Who was fighting and who/what was being fought? The construction of participants’ identities in UK and US reporting of the Iraq War. In Garzone, Giuliana & Catenaccio, Paola, eds., *Identities across Media and Modes: Discursive Perspectives*. Bern: Peter Lang, pp. 259–87.
- Matley, David. (2017). This is NOT a #humblebrag, this is just a #brag: The pragmatics of self-praise, hashtags and politeness in Instagram posts. *Discourse, Context & Media* 22, 30–8.
- Maynard, Carson & Leicher, Sheryl. (2007). Pragmatic annotation of an academic spoken corpus for pedagogical purposes. In Fitzpatrick, Eileen, ed., *Corpus Linguistics beyond the Word: Corpus Research from Phrase to Discourse*. Amsterdam: Brill, pp. 107–15.
- McEnery, Tony, Xiao, Richard & Tono, Yukio. (2005). *Corpus-Based Language Studies*. London: Routledge.
- Méndez-Naya, Belén & Pahta, Päivi. (2010). Intensifiers in competition: The picture from early English medical writing. In Taavitsainen, Irma & Pahta, Päivi, eds., *Early Modern English Medical Texts: Corpus Description and Studies*. Amsterdam/Philadelphia: John Benjamins, pp. 191–213.
- Messerli, Thomas C. (2020). Ocean’s Eleven scene 12 – Lost in transcription. *Perspectives* 28(6), 837–50. <https://doi.org/10.1080/0907676X.2019.1708421>.

- Messerli, Thomas C., & Locher, Miriam A. (2021). Humour support and emotive stance in comments on K-Drama. *Journal of Pragmatics* 178, 408–25. <https://doi.org/10.1016/j.pragma.2021.03.001>.
- Miller, John, & Gergen, Kenneth J. (1998). Life on the line: The therapeutic potentials of computer-mediated conversation. *Journal of Marital and Family Therapy* 24(2), 189–202. <https://doi.org/10.1111/j.1752-0606.1998.tb01075.x>.
- Morrow, Phillip R. (2012). Online advice in Japanese: Giving advice in an Internet discussion forum. In Limberg, Holger & Locher, Miriam A., eds., *Advice in Discourse*. Amsterdam: John Benjamins, pp. 255–79.
- Norrick, Neal R. (2009). Interjections as pragmatic markers. *Journal of Pragmatics* 41(5), 866–91.
- O’Keeffe, Anne. (2018). Corpus-based function-to-form approaches. In Jucker, Andreas H., Schneider, Klaus P. & Bublitz, Wolfram, eds., *Methods in Pragmatics* (Handbooks of Pragmatics 10). Berlin: De Gruyter Mouton, pp. 587–618.
- O’Keeffe, Anne, Clancy, Brian & Adolphs, Svenja. (2020). *Introducing Pragmatics in Use*. 2nd ed. London: Routledge.
- Palander-Collin, Minna. (1999). *Grammaticalization and Social Embedding. I THINK and METHINKS in Middle and Early Modern English*. Helsinki: Société Néophilologique.
- Placencia, María Elena. (2012). Online peer-to-peer advice in Spanish Yahoo! Respuestas. In Limberg, Holger & Locher, Miriam A., eds., *Advice in Discourse*. Amsterdam: John Benjamins, pp. 281–305.
- Proferes, Nicholas, Jones, Naiyan, Gilbert, Sarah, Fiesler, Casey & Zimmer, Michael. (2021). Studying Reddit: A systematic overview of disciplines, approaches, methods, and ethics. *Social Media + Society* 7(2), 1–14. <https://doi.org/10.1177/20563051211019004>.
- Quo VaDis Project. (2022). Questioning vaccination discourse. [www.lancaster.ac.uk/vaccination-discourse](http://www.lancaster.ac.uk/vaccination-discourse).
- Ranganath, Rajesh, Jurafsky, Dan, & McFarland, Dan. (2009). It’s not you, it’s me: Detecting flirting and its misperception in speed-dates. In Koehn, Philipp & Mihalcea, Rada, eds., *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (Vol. 1). Morristown, NJ: Association for Computational Linguistics, pp. 334–42.
- Rebora, Simone, Boot, Peter, Pianzola, Federico, et al. (2021). Digital humanities and digital social reading. *Digital Scholarship in the Humanities* 36(S2), ii230–50. <https://doi.org/10.1093/lhc/fqab020>.
- Romero-Trillo, Jesús, ed. (2008). *Pragmatics and Corpus Linguistics: A Mutualistic Entente*. Berlin: Mouton de Gruyter.

- Rudolf von Rohr, Marie-Thérèse. (2015). "You will be glad you hung onto this quit": Sharing information and giving support when stopping smoking online. In Smith, Catherine Arnott, & Keselman, Alla, eds., *Meeting Health Information Needs outside of Healthcare: Opportunities and Challenges*. Waltham, MA: Chandos/Elsevier, pp. 263–90.
- Rudolf von Rohr, Marie-Thérèse. (2017). "If you start again, don't worry. You haven't failed": Relapse talk and motivation in online smoking cessation. *Linguistics Online* 87(8/17), 87–105. <http://dx.doi.org/10.13092/lo.87.4174>.
- Rudolf von Rohr, Marie-Thérèse. (2018). Persuasion in smoking cessation online: An interpersonal pragmatic perspective. Freiburg i. Br.: Albrecht-Ludwigs-Universität Freiburg / Universitätsbibliothek Freiburg, [https://freidok.uni-freiburg.de/fedora/objects/freidok:16755/datastreams\(FILE1/content](https://freidok.uni-freiburg.de/fedora/objects/freidok:16755/datastreams(FILE1/content)
- Rudolf von Rohr, Marie-Thérèse, & Locher, Miriam A. (2020). The interpersonal effects of complimenting others and self-praise in online health settings. In Placencia, Mária Elena & Eslami, Zohreh Rasekh, eds., *Complimenting Behavior and (Self-)Praise across Social Media*. Amsterdam: John Benjamins, pp.189–211.
- Rudolf von Rohr, Marie-Thérèse, Thurnherr, Franziska & Locher, Miriam A. (2019). Linguistic expert creation in online health practices. In Bou-Franch, Patricia & Garcés-Conejos Blitvich, Pilar, eds., *Analysing Digital Discourse: New Insights and Future Directions*. London: Palgrave Macmillan, pp.219–50.
- Rühlemann, Christopher. (2019). *Corpus Linguistics for Pragmatics: A Guide for Research*. Oxon: Routledge.
- Rühlemann, Christopher & Aijmer, Karin. (2014). Corpus pragmatics: Laying the foundations. In Aijmer, Karin & Rühlemann, Christoph, eds., *Corpus Pragmatics: A Handbook*. Cambridge: Cambridge University Press, pp. 1–26.
- Rühlemann, Christopher & Clancy, Brian. (2018). Corpus linguistics and pragmatics. In Ilie, Cornelia & Norrick, Neal R., eds., *Pragmatics and its Interfaces*. Amsterdam: John Benjamins, pp. 241–66.
- Saldaña, Johnny. (2013). *The Coding Manual for Qualitative Researchers* (2nd ed.). London: SAGE Publications.
- Simon-Vandenbergen, Anne-Marie. (2000). The functions of *I think* in political discourse. *International Journal of Applied Linguistics* 10(1), 41–63.
- Soffritti, Marcello. (2018). Multimodal corpora in audiovisual translation studies. In Pérez-González, Luis, ed., *The Routledge Handbook of Audiovisual Translation* (1st ed.). London: Routledge, pp. 334–49.
- Sotillo, Susana. (2012). Illocutionary acts and functional orientation of SMS texting in SMS social networks. In Ebeling, Signe Oksefjell, Ebeling, Jarle & Hasselgård, Hilde, eds., *Aspects of Corpus Linguistics: Compilation, Annotation, Analysis*

- (article 10). Helsinki: VARIENG, <https://varieng.helsinki.fi/series/volumes/12/index.html>.
- Suhr, Carla & Taavitsainen, Irma, eds. (2012). *Developing Corpus Methodology for Historical Pragmatics* (Studies in Variation, Contacts and Change in English 11). Helsinki: VARIENG. <https://varieng.helsinki.fi/series/volumes/11/>.
- Taavitsainen, Irma. (2001). Evidentiality and scientific thought-styles: English medical writing in Late Middle English and Early Modern English. In Gotti, Maurizio & Dossena, Marina, eds., *Modality in Specialized Texts*. Bern: Peter Lang, pp. 21–52.
- Taavitsainen, Irma. (2002). Historical discourse analysis: Scientific language and changing thought-styles. In Fanego, Teresa, Méndez-Naya, Belén & Seoane, Elena, eds., *Sounds, Words, Texts and Change: Selected Papers from 11 ICEHL, Santiago de Compostela, 7-11 September 2000*. Amsterdam: John Benjamins, pp. 201–26.
- Taavitsainen, Irma. (2009). The pragmatics of knowledge and meaning: Corpus linguistic approaches to changing thought-styles in early modern medical discourse. In Jucker, Andreas H., Schreier, Daniel & Hundt, Marianne, eds., *Corpora: Pragmatics and Discourse. Papers from the 29th International Conference on English Language Research on Computerized Corpora (ICAME 29). Ascona, Switzerland, 14-18 May 2008*. Amsterdam: Rodopi, pp. 37–62.
- Taavitsainen, Irma. (2016). Genre dynamics in the history of English. In Kyöö, Merja & Pahta, Päivi, eds., *The Cambridge Handbook of English Historical Linguistics*. Cambridge: Cambridge University Press, pp. 271–85.
- Taavitsainen, Irma, Jucker, Andreas H. & Tuominen, Jukka, eds. (2014). *Diachronic Corpus Pragmatics* (Pragmatics & Beyond New Series 243). Amsterdam: John Benjamins.
- Thurnherr, Franziska. (2017). “As it’s our last exchange next time...”. The closure initiation in email counseling. *Linguistics Online* 87(8), 213–36. <http://dx.doi.org/10.13092/lo.87.4180>.
- Thurnherr, Franziska. (2022). *Relational Work and Identity Construction in Email Counseling*. Freiburg im Breisgau: Albrecht-Ludwigs-Universität Freiburg / Universitätsbibliothek Freiburg.
- Thurnherr, Franziska, Rudolf von Rohr, Marie-Thérèse & Locher, Miriam A. (2016). The functions of narrative passages in three written online health contexts. *Open Linguistics* 2(1), 450–70. <https://doi.org/10.1515/opli-2016-0024>.
- Tiedemann, Jörg. (2012). Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and*

- Evaluation (LREC'2012). [www.lrec-conf.org/proceedings/lrec2012/pdf/463\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf).
- Tognini-Bonelli, Elena. (2001). *Corpus Linguistics at Work*. Amsterdam: John Benjamins.
- Tonetti Tübben, Ilenia & Landert, Daniela. (2022). *Uh and um as pragmatic markers in dialogues: A contrastive perspective on the functions of planners in fiction and conversation*. *Contrastive Pragmatics* 1–32. <https://doi.org/10.1163/26660393-bja10049>.
- Tottie, Gunnell. (2011). *Uh and um as sociolinguistic markers in British English*. *International Journal of Corpus Linguistics* 16(2), 173–97.
- Tottie, Gunnell. (2014). On the use of *uh* and *um* in American English. *Functions of Language* 21(1), 6–29.
- Tottie, Gunnell. (2019). From pause to word: *Uh, um* and *er* in written American English. *English Language and Linguistics* 23(1), 105–30.
- Traugott, Elizabeth Closs. (2015). “Ah, pox o’ your pad-lock”: Interjections in the Old Bailey Corpus 1720–1913. *Journal of Pragmatics* 86, 68–73.
- Tyrkkö, Jukka, Hickey, Raymond & Marttila, Ville. (2010). Exploring Early Modern English medical texts. Manual for EMEMT Presenter. In Taavitsainen, Irma & Pahta, Päivi, eds., *Early Modern English Medical Texts: Corpus Description and Studies*. Amsterdam: John Benjamins, pp. 219–77.
- Vaughan, Elaine, McCarthy, Michael & Clancy, Brian. (2017). Vague category markers as turn-final items in Irish English. *World Englishes* 36(2), 208–23.
- Virtanen, Tuija & Halmari, Helena, eds. (2005). *Persuasion across Genres*. Amsterdam: John Benjamins.
- Weisser, Martin. (2010). Annotating dialogue corpora semi-automatically: A corpus-linguistic approach to pragmatics. Habilitation (professorial) thesis, University of Bayreuth.
- Weisser, Martin. (2014). Manual for the Dialogue Annotation & Research Tool (DART). [http://martinweisser.org/publications/DART\\_manual.pdf](http://martinweisser.org/publications/DART_manual.pdf).
- Weisser, Martin. (2015). Speech act annotation. In Aijmer, Karin & Rühlemann, Christopher, eds., *Corpus Pragmatics: A Handbook*. Cambridge: Cambridge University Press, pp. 84–114.
- Weisser, Martin. (2018). *How to Do Corpus Pragmatics on Pragmatically Annotated Data: Speech Acts and Beyond*. Amsterdam: John Benjamins.
- Whitt, Richard J. (2016). Using corpora to track changing thought styles: Evidentiality, epistemology, and Early Modern English and German scientific discourse. *Kalbotyra* 69, 265–91.
- Zufferey, Sandrine & Cartoni, Bruno. (2014). A multifactorial analysis of explication in translation. *Target* 26(3), 361–84.

## Acknowledgements

Daniela's work on this Element was conducted at the University of Basel as well as at Heidelberg University. Miriam wishes to thank the University of Basel for financing a research leave in spring 2022 and Korea University in Seoul for hosting her during the writing period of this book. We thank the anonymous reviewers and general editors for their constructive feedback, which helped to strengthen our line of argumentation. This open access manuscript has been published with the support of the Swiss National Science Foundation.

# **Funding Statement**

Published with the support of the Swiss National Science Foundation.

## Pragmatics

---

Jonathan Culpeper

Lancaster University, UK

Jonathan Culpeper is Professor of English Language and Linguistics in the Department of Linguistics and English Language at Lancaster University, UK. A former co-editor-in-chief of the *Journal of Pragmatics* (2009–14), with research spanning multiple areas within pragmatics, his major publications include: *Impoliteness: Using Language to Cause Offence* (2011, CUP) and *Pragmatics and the English Language* (2014, Palgrave; with Michael Haugh).

Michael Haugh

University of Queensland, Australia

Michael Haugh is Professor of Linguistics and Applied Linguistics in the School of Languages and Cultures at the University of Queensland, Australia. A former co-editor-in-chief of the *Journal of Pragmatics* (2015–2020), with research spanning multiple areas within pragmatics, his major publications include: *Understanding Politeness* (2013, CUP; with Dániel Kádár), *Pragmatics and the English Language* (2014, Palgrave; with Jonathan Culpeper), and *Im/politeness Implicatures* (2015, Mouton de Gruyter).

### Advisory Board

Anne Baron Leuphana University of Lüneburg, Germany

Betty Birner Northern Illinois University, USA

Lucien Brown Monash University, Australia

Billy Clark Northumbria University, UK

Chris Cummins University of Edinburgh, UK

Pilar Garcés-Conejos Blitvich University of North Carolina at Charlotte, USA

Andreas H. Jucker University of Zurich, Switzerland

Zohar Kampf Hebrew University of Jerusalem, Israel

Miriam A. Locher University of Basel, Switzerland

Yoshiko Matsumoto Stanford University, USA

Marina Terkourafi Leiden University, The Netherlands

Chaoqun Xie Zhejiang International Studies University, China

---

### About the series

*Cambridge Elements in Pragmatics* showcases dynamic and high-quality original, concise and accessible scholarly works. Written for a broad pragmatics readership, it encourages dialogue across different perspectives on language use. It is a forum for cutting-edge work in pragmatics: consolidating theory, leading the development of new methods, and advancing innovative topics in the field.

## Pragmatics

---

### Elements in the Series

*Advice in Conversation*

Nele Pöldvere, Rachele De Felice and Carita Paradis

*Positive Social Acts: The Brighter and Darker Sides of Sociability*

Roni Danziger

*Pragmatics in Translation: Mediality, Participation and Relational Work*

Daria Dayter, Miriam A. Locher and Thomas C. Messerli

*Fiction and Pragmatics*

Miriam A. Locher, Andreas H. Jucker, Daniela Landert and Thomas C. Messerli

*Corpus Pragmatics*

Daniela Landert, Daria Dayter, Thomas C. Messerli and Miriam A. Locher

A full series listing is available at: [www.cambridge.org/EIPR](http://www.cambridge.org/EIPR)

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/336055983>

# Mining user interaction patterns in the darkweb to predict enterprise cyber incidents

Preprint · September 2019

---

CITATIONS

0

READS

135

4 authors:



Soumajyoti Sarkar  
Amazon  
31 PUBLICATIONS 115 CITATIONS

[SEE PROFILE](#)



Mohammed Almukaynizi  
Arizona State University  
18 PUBLICATIONS 208 CITATIONS

[SEE PROFILE](#)



Jana Shakarian  
self  
43 PUBLICATIONS 497 CITATIONS

[SEE PROFILE](#)



Paulo Shakarian  
Arizona State University  
187 PUBLICATIONS 2,113 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Strongly Hierarchical Factorization Machines and ANOVA Kernel Regression [View project](#)



Cyber-Attack Prediction [View project](#)

---

## Mining user interaction patterns in the darkweb to predict enterprise cyber incidents

Soumajyoti Sarkar · Mohammad Almukaynizi · Jana Shakarian · Paulo Shakarian

Received: date / Accepted: date

**Abstract** With rise in security breaches over the past few years, there has been an increasing need to mine insights from social media platforms to raise alerts of possible attacks in an attempt to defend conflict during competition. In this study, we attempt to build a framework that utilizes unconventional signals from the darkweb forums by leveraging the reply network structure of user interactions with the goal of predicting enterprise related external cyber attacks. We use both unsupervised and supervised learning models that address the challenges that come with the lack of enterprise attack metadata for ground truth validation as well as insufficient data for training the models. We validate our models on a binary classification problem that attempts to predict cyber attacks on a daily basis for an organization. Using several controlled studies on features leveraging the network structure, we measure the extent to which the indicators from the darkweb forums can be successfully used to predict attacks. We use information from 53 forums in the darkweb over a span of 17 months for the task. Our framework to predict real world organization cyber attacks of 3 different security events, suggest that focusing on the reply path structure between groups of users based on random walk transitions and community structures has an advantage in terms of better performance solely relying on forum or user posting statistics prior to attacks.

**Keywords** Social Networks · Cyber attack prediction · Machine Learning

---

S. Sarkar, M. Almukaynizi, P. Shakarian  
Arizona State University  
E-mail: ssarka18@asu.edu  
E-mail: malmukay@asu.edu  
E-mail: shak@asu.edu

J. Shakarian  
Cyber Reconnaissance Inc.  
E-mail: jana@cyr3con.ai

## 1 Introduction

With recent data breaches such as those of Yahoo, Uber, Equifax<sup>1</sup> among several others that emphasize the increasing financial and social impact of cyber attacks, there has been an enormous requirement for technologies that could provide such organizations with prior alerts on such data breach possibilities. Such security threat intelligence information would help address the following: (1) while organizations spend a lot of money to secure network systems that could avoid such data breaches [1], it is not devoid of exposures to vulnerabilities specially as such platforms depend on a large number of third party software systems. (2) an alert for a possible intrusion into technology platforms like email servers or malware injection into softwares could actually help organizations focus on a specific set of components in a short time, thereby allowing faster security tightening to avoid being exploited on a regular basis [2].

The total number of data breaches in 2017 crossed 1000<sup>2</sup> across all sectors which is a record high, considering previous years and exposing over a billion records containing sensitive data. On the vulnerability front, the Risk Based Security's VulnDB database<sup>3</sup> published a total of 4837 vulnerabilities in a quarter of 2017 which was around 30% higher than previous year. This motivates the need for an extensive application that can track vulnerability based information from external sources to raise alerts on such data breaches. While the darkweb is one such place on the internet where users can share information on software vulnerabilities and ways to exploit them [3,4] and where it might be difficult to track the actual identity of those users, what they leave behind are the footprints of their posting and interaction patterns in forums. In this paper, as one of contributions in the field of cyber attack prediction, we leverage the information obtained from evolving reply networks of discussions in the darkweb forums while also capturing the user and thread posting statistics in these forums to understand the extent to which the darkweb information can be useful as signals for predicting real world target specific enterprise cyber attacks.

In the vulnerability lifecycle, a vulnerability goes through multiple stages. It starts with being undisclosed when the general public does not know about it and attackers can identify them, develop exploits and use them for “zero-day” attacks. However, once a vulnerability is identified, an indexing is done with an ID assigned to it - that is where the vendor starts working on a patch. Once a patch is released, is when hackers would try to reverse engineer the patch and develop exploits. The last stage would generally entail using metasploit modules to launch these attacks via these exploits [5]. So in this vulnerability lifecycle, the significance of discussions in darkweb forums or most social media platforms can appear in two phases: once when the vulnerability is undisclosed

---

<sup>1</sup> <https://www.consumer.ftc.gov/blog/2017/09/equifax-data-breach-what-do>,  
<https://www.consumer.ftc.gov/blog/2016/09/yahoo-breach-watch>

<sup>2</sup> <https://www.wombatsecurity.com/blog/scary-data-breach-statistics-of-2017>

<sup>3</sup> <https://www.riskbasedsecurity.com/2017/05/29/increase-in-vulnerabilities-already-disclosed-in-2017/>

---

and there are discussions revolving around related vulnerabilities or exploits and second, after the patch is released but before an exploit is materialized for an attack. So our goal is to leverage the discussions in these two phases to be able to predict cyber attacks before the exploit is weaponized.

We attempt to build an integrated approach utilizing unconventional signals from the darkweb discussions for predicting attacks on a target organization - here “unconventional” means that the information from the darkweb might not necessarily be observables of the actual attacks on the target organization. This is in contrary to traditional studies where authors use system level features within the target organization to predict attacks in future for the same or related organization [23,8]. With this in mind, we hypothesize that the interaction dynamics focused on a set of specialized users and the attention broadcast by them to other posts in these underground platforms can be one of ways to generate warnings for future attacks. We mine patterns of anomalous behavior from these forums and use them directly for cyber attack prediction on external enterprise host systems. We note that we *do not* consider whether vulnerabilities mentioned in forum discussions, have been exploited or not as the basis for attacks since a lot of zero day attacks [9] might occur before such vulnerabilities are even indexed and their gravity might lie hidden in discussions related to other associated vulnerabilities or some discussion on exploits. The premise on which this research is setup is based on evaluating the dynamics of all kinds of discussions in the darkweb forums but we attempt to filter out the noise to mine important patterns by studying whether a piece of information gains traction within important communities. So in this sense we do not explicitly focus on discussions relating vulnerabilities exclusively nor their exploits to predict real world cyber attacks.

We try to quantify the correlation between the pattern of replies by a specific group of users we term *experts* who engage more frequently with popular vulnerability mentions in their posts over time and which gain attention from other users, and a real world cyber attack in the near future as a first challenge in this study. A second research opportunity in this direction is to see whether we can use company agnostic unsupervised models that overcome the lack of company specific metadata from the attack ground truth. We investigate the extent to which we can correlate anomalies from these darkweb network interactions to near term cyber attacks and how well they materialize for different companies. To this end, the major contributions of this research investigation are as follows:

- We create a novel network mining technique using the directed reply network of users to extract a set of specialized users we term *experts* whose posts with popular vulnerability mentions gain attention from other users in a specific time frame. Following this, we generate several time series of features that capture the dynamics of interactions centered around these *experts* across individual forums of the darkweb.
- We apply a widely used unsupervised anomaly detection technique that uses residual analysis to detect anomalies and propose an anomaly based

attack prediction technique on a daily basis. Additionally, we also train a supervised learning model based on logistic regression with attack labels from an organization to predict daily attacks.

- Empirical evidence from our unsupervised anomaly detector suggests that a feature based on graph conductance that measures the random walk transition probability between groups of users is a useful indicator for attack occurrences given that it achieved the best AUC score of 0.69 for one type of attacks. We obtain similar best results for the supervised model having the best F1 score of 0.53 for the same feature and attack type compared to the random (without prior probabilities) F1 score of 0.37. We additionally investigate the performance of the models in weeks where frequency of attacks are higher and find the superior performance of community structures in networks in predicting these attacks.

To the best of our knowledge, this is a first attempt in creating a framework that investigates the network structure of the darkweb forums data as an external source of information to generate alerts and predict real world cyber attacks without having the need to monitor vulnerability prioritization or exploitation.

## 2 Related Work and Motivation

In this work, we discuss some of the past and ongoing research in the domain of cyber security analytics that also caters to the general area of predicting future cyber breach incidents in real world systems. Most of the work on vulnerability discussions on trading, exploitation in the underground forums [10, 11, 28] and related social media platforms like Twitter[14, 13, 12] have focused on two aspects: (1) analyzing the dynamics of the underground forums and the markets that drive it, thereby focusing on mechanisms that enable the market activity, and giving rise to the belief that the “lifecycle of vulnerabilities” in these forums and marketplaces have significant impact on real world cyber attacks [21, 9] (2) prioritization of vulnerabilities using these social media platforms or binary file appearance logs of machines and using them to predict the risk state of machines or systems through exploitation of these vulnerabilities [8]. So, the two components in majority of these studies that have been repeatedly worked upon in silos are analysis of vulnerabilities and their likelihood of exploitation in these forums or platforms and, then vulnerability exploitation severity based prediction to associate them to real world cyber breach incidents [5, 12]. In this paper, we ignore the gap between vulnerability exploit analysis and the final task of real world cyber attack prediction by removing the preconceived notions used in earlier studies where vulnerability exploitation is considered a precursor towards attack prediction.

The rapid expansion of the cyber-threat landscape is augmented by the presence of underground platforms that support the discussion, proliferation of exploit awareness, deployment and monetization of such exploits leading

Symbols	Definition
$f, F$	a particular (single) forum, set of forums considered in study
$t$	discrete time point instance
$A$	set of attack types: malicious-email, endpoint malware and malware destination
$\Gamma$	global time range of points in our study $\{t_1, t_2 \dots t_k\}$
$\tau$	an ordered subsequence of time points $\in \Gamma$
$x$	feature in our study of machine learning based prediction models
$h$	a thread in a forum ( a thread is a series of posts on a particular topic initiated by a user)
$p_{h,i}$	in a chronologically ordered set of posts in thread $h$ , it denotes the $i^{th}$ post (from beginning) in $h$
$\mathcal{T}_{x,f}$	a time series data for feature $x$ from discussion posts in forum $f$
$\mathcal{R}_x$	residual vector time series data for feature $x$ from discussion posts aggregated over all forums
$H_\tau$	historical time period (prior to $\tau$ ) w.r.t. $\tau$ , $\forall t' \in H_\tau$ and for any $t \in \tau$ , $t' < t$ and there is a time gap between the start of $\tau$ and end of $H_\tau$
$V_\tau^f, E_\tau^f$	set of nodes in reply network from forum $f$ discussions and in time period $\tau$ , set of edges with the same constraints (we drop $f$ when we generalize for all forums)
$G_{H_\tau}$	Reply network induced by discussion in $H_\tau$
$exp_\tau$	Experts in the time subsequence $\tau$
$\mathbf{X}$	#features $\times$ T matrix (T denotes the time dimension)
$\mathbf{Y}$	T $\times$ F matrix
$\mathbf{y}$	1 $\times$ F vector
$\beta$	weight for a feature in the logistic regression model
$\eta$	time window for feature selection in $\mathcal{T}_{x,f}$
$\delta$	time gap between attack prediction at a time point and the feature window
$\zeta$	anomaly to attack prediction (anomaly) count threshold parameter

Table 1: Table of notations

to cyber-attacks [15, 16, 17, 10]. However, despite the existing literature that studies the economies of these underground forums and markets present in the darkweb, there has been very few studies that focus on filtering the markets and forums that actually contribute to the threat scenario [18, 19, 20]. One of the ways to understand the indicators surrounding these underground platforms, that could lead to potentially malicious attempts to breach systems at scale is to monitor the interactions that receive attention in these platforms.

We discuss 3 areas within which our work falls when we discuss the landscape of cyber attack prediction based on signals from social media and attacks on an organization. However, we point out the main differences that bring out the significance and novelty of our approach and the problem we attempt to solve in the following:

1. *Cyber attack with within-organization system signals:* Cyber attack prediction on external organizations have recently been studied in the context of feature engineering for gathering predictive signals. Some of the most related works in this area include a study [1], where features are gathered from the network systems and the log files of a target organization. These

features are then used for training a classifier to predict future attacks for the same organization, and where the ground truth for the attacks are collected from reported cyber incidents from Web Hacking Database, Hackmageddon. Contrary to this, we use unconventional signals from the darkweb that are not necessarily observables of the attacks for the organization but we try to measure the extent to which they can perform well over other measures. Similar to this study, there have already been attempts to develop systems at scale that could predict the risk of systems by analyzing various sensors such as binary appearance of log files [8].

2. *Cyber attack prediction using social media data:* There have been several attempts to use external social media data sources to predict real world cyber attacks [1, 23, 13, 22]. However, the problem these studies focus on is to build predictive models to correlate the social media signals to attacks in the real world that are not observed for a specific organization. Our attack prediction problem specifically proposes to build models specific to an external organization using external sensors not obtained from the internal system data for the same organization. One of the closest works in this area is done by authors in [31], where the authors use signals using GDELT, Twitter and OTX based on keywords related to the organization. One of the challenges related to our dataset is that we did not find any keywords directly related to the name of our target organization in the darkweb - similar issues are reported in [30] where the authors relied on some curated keyword search from Twitter and blogs and the darkweb for attack prediction. Our work has a slight advantage in that our selection of forums and the features including vulnerability information does not depend on human engineered knowledge, rather it focuses on the trends in time - so in a sense our streaming nature of prediction is scalable.
3. *Social Network analysis for cyber security:* Using network analysis to understand the topology of darkweb forums has been studied at breadth in [24] where the authors use social network analysis techniques on the reply networks of forums in order to identify members of Islamic community within the darkweb. Similarly in [25], the authors use topic modeling and the network structure of the darkweb forums in order to understand the interactions between extremist groups. However, such analysis of reply networks have been conducted on static networks [6] where authors devised network features of users for predictive modeling. A recent study done in [26] shows how to leverage the network structure of these reply networks for cyber attack prediction. These studies suggest that the nature of interactions can unveil important actors in darkweb forums and their activity regarding discussions can provide us with signals for cyber attacks. One of our contributions in this paper is that we use evolving networks of the users with certain constraints that can now be leveraged for streaming prediction on a daily basis in an automated manner. Our hypothesis lies on the premise that the attention broadcast by these users towards other posts

are in fact sensors for impending cyber attacks. Such studies of separating specialized users have been studied before in the context of trading financial information in carding forums [27].

The rest of the paper is organized as follows: we first introduce a few security terminologies relevant to our work and the dataset sources and attributes in Section 3, following which we formally define the prediction problem attempted in this paper in Section 4. We then discuss the technical details of our attack prediction framework including the feature engineering and the model learning components in Section 5. We discuss the experimental settings and the results in Section 6 and finally we end this work with some discussion and case studies in Section 7.

### 3 Cyber Security terms and Dataset

We first introduce a few terms commonly used in the cyber security domain and that we would use in this paper frequently. Vulnerability is a weakness in a software system that can be exploited by an attacker to compromise the confidentiality, integrity or availability of the system to cause harm [29].

*Common Vulnerabilities and Exposures (CVE)* : The database of Common Vulnerabilities and Exposures maintained on a platform operated by the MITRE corporation<sup>4</sup> provides an identity mapping for publicly known information-security vulnerabilities and exposures.

*Common Platform Enumeration (CPE)*: A CPE is a structured naming scheme for identifying and grouping clusters of information technology systems, software and packages maintained in a platform NVD (National Vulnerability Database) operated by NIST<sup>5</sup>.

*CVE - CPE mapping*: Each CVE can be assigned to different CPE groups based on the naming system of CPE families as described in [6]. Similarly, each CPE family can have several CVEs that conform to its vendors and products that the specific CPE caters to. For the purpose of this paper, we form a simplified grouping hierarchy to cluster the CVEs by their CPE levels which we describe in Section 3.3.

*Forum topic*: Each darkweb forum or site  $f$  consists of several threads  $h$  initiated by a specific user and over time, several users post and reply in these threads. We note that one user can appear multiple times in the sequence of posts depending on when and how many times the user posted in that thread. Since each thread is associated with a topic (or a title), we would often use the terms topic to refer to a particular thread  $h$  comprising all posts in the relevant forum. We denote the set of these 53 forums used in this dataset using the symbol  $F$ .

The ground truth and the darkweb data have been collected from two different sources as will be described in the following sections and although we

---

<sup>4</sup> <https://www.mitre.org/>

<sup>5</sup> <https://www.nist.gov/>

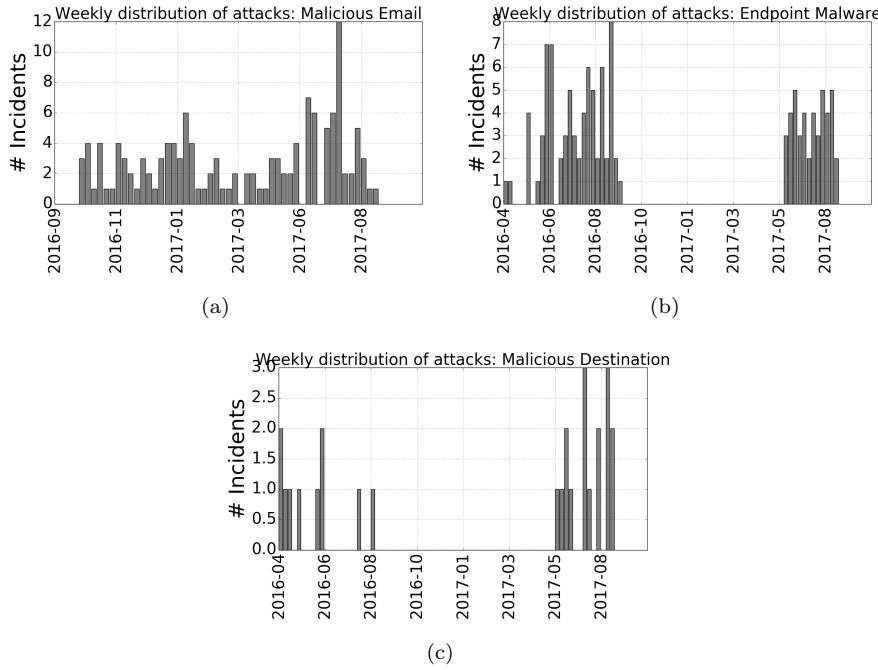


Fig. 1: Weekly occurrence of security breach incidents of different types (a) Malicious email (b) Endpoint Malware (c) Malicious destination

validate our prediction models based on the available ground truth, we perform extensive case studies to show the significance of our prediction models in the real world.

### 3.1 Enterprise-Relevant External Threats (GT)

We use the cyber attacks Ground Truth (GT) from the data provided from a corporate entity to funders of this work<sup>6</sup>. The corporate entity is *Armstrong Corporation* to conceal the actual identity. The data contains information on cyber attacks on their systems in the period of April 2016 to September 2017. Each data point is a record of a detected deliberate malicious attempt to gain unauthorized access, alter or destroy data, or interrupt services or resources in the environment of the participating organization. Those malicious attempts were real-world events detected in the wild, in uncontrolled environment, and by different attack detectors such as anti-virus and IDS software and hardware products. The data contains the following relevant attributes: { *event-type*: The type of attack which are categorized as malicious-email, endpoint-malware and malicious-destination, *event occurred date*: Date on which there

<sup>6</sup> <https://www.iarpa.gov/index.php/research-programs/cause>

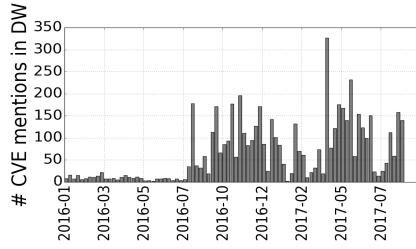
was an attack of particular event-type, *event reported date*: Date on which the attack was reported, *detector*: the software service that detected the system intrusion attempting to break into their systems, *threat\_designation\_family*: the categories of threats from among a Threat Family Dictionary. }. The *event-types* that are used in this study are:

- *Malicious Email*: A malicious attempt is identified as a Malicious Email event if an email is received by the organization, and it either contains a malicious email attachment, or a link (embedded URL or IP address) to a known malicious destination.
- *Malicious Destination*: A malicious attempt is identified as visit to a Malicious Destination if the visited URL or IP address hosts malicious content.
- *Endpoint Malware*: A Malware on Endpoint event is identified if malware is discovered on an endpoint device. This includes, but not limited to, ransomware, spyware, and adware.

We denote these set of attack types as *A*. Here the term “malicious” means that the end goal of all these 3 attempts were to intrude the systems of the host enterprise and exploit them, however to what extent are they successful is not known and is not a matter of concern for an incident to be qualified as a cyber-attack. In our research, we use the categories: *event-type* and *attack occurred date* as our ground truth (GT) for validation and avoid the use of other attributes present in the dataset as they are metadata provided by third party software services which are not available for all security incident reports. Additionally, since our research is focused on using the darkweb as an external source of data to capture the behavioral patterns of user interactions, we only use the *event-type* and *event-occurred-date* as our ground truth. We note that the absence of information that can accurately provide us with information regarding vulnerabilities and exploits that caused the attacks, for our model validation makes the problem more challenging. As shown in Figure 1, the distribution of attacks over time is different for the 3 events. Additionally, we also observe that for the events *endpoint-malware* shown in Figure 1(b) and *malicious-destination* shown in Figure 1(c), the weekly occurrence has not been captured consistently and there is missing information for these events in few time intervals. We take note of this while building our learning models to predict the occurrence of an attack. The total number of incidents reported for the events are as follows: 26 incidents tagged as *malicious-destination*, 119 tagged as *endpoint-malware* and 135 for *malicious-email* events resulting in a total of 280 incidents over a span of 17 months that were considered in our study.

### 3.2 Darkweb data

The entire focus of this research has been to disentangle the interactions centered around a few users over time and the noise that is present in the form of random discussions in different forums. It helps us at assessing whether



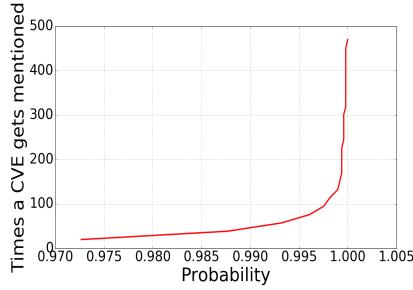


Fig. 3: Cumulative Distribution Function (cdf) showing the number of times each CVE is mentioned in posts in the darkweb.

vulnerabilities without looking at the content or the user-user network is very difficult.

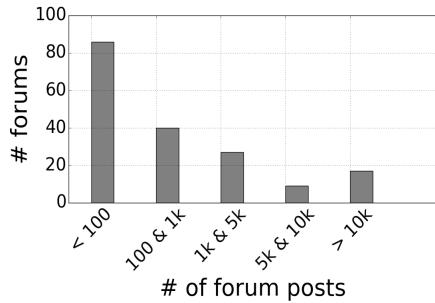


Fig. 4: Distribution of the number of forums posts across all forums

*Forums data:* In this paper, we consider the dynamics of interactions in dark-web forums and for that we filter out forums based on a threshold number of posts that were created in the timeframe of January 2016 to September 2017. We gathered data from 179 forums in that time period where the total number of unique posts irrespective of the thread that they belonged to, were 557689. As shown in Figure 4, the number of forums with less than 100 posts is large and therefore we only consider forums which have greater than 5000 posts in that time period which gave us a total of 53 forums. As will be described later, we rely on a projection method to compute lower dimensional features and hence any significant patterns occurring out of these forums would be captured without the requirement to manually filter and select particular forums. We note that unlike some related research using darkweb for cyber attack prediction which use large number of forums for obtaining signals for prediction [30], we refrain from using forums with not enough data in the 1-year period of our study. This is to avoid the issues of missing data on days where we would

need to predict attacks - an imputation measure for this is an active area of research [31] and we consider this as a step towards our future work.

### 3.3 CPE Groups

We gather the CPE data for all the vulnerabilities relevant to the darkweb discussions in our study from the publicly available repository of CPE data. In order to cluster the set of CVEs into a set of CPE groups, we use the set of CPE tags for each CVE from the NVD database maintained by NIST. For the CPE tags, we only consider the operating system platform and the application environment tags for each unique CPE. Examples of CPE would include: *Microsoft Windows\_95*, *Canonical ubuntu\_linux*, *Hp elitebook\_725\_g3*. The first component in each of these CPEs denote the operating system platform and the second component denotes the application environment and their versions. Some of the CPE groups might be a parent cluster of another CPE group. For example, *Microsoft Windows* would be a parent cluster for CPEs like *Microsoft Windows\_8* or *Microsoft Windows\_10*. In this research, we do not consider any hierarchies in the CPEs for filtering out clusters, but as future research use, this can be considered. From our data we found that over the time period from April 2016 to September 2017, the top CPE groups having CVEs which are mentioned most widely in darkweb forum posts are ntp, php, adobe flash\_player, microsoft windows\_server\_2008, linux kernel, microsoft windows\_7, micorosft windows\_server\_2012, and canonical ubuntu\_linux.

## 4 Prediction Problem

Before we describe our framework for using darkweb discussions in the forums for predicting external enterprise attacks, we formally describe our prediction task. Formally, given a target organization  $E$ , a set of unconventional (external) signals from the darkweb forums as features and a set of  $A$  attack types for  $E$ , we solve a binary classification problem that investigates whether there would be an attack (0/1) of any type in  $A$  for  $E$  on a daily basis.

The mechanism for attack predictions as shown in Figure 5 can be described in 3 steps : (1) given a time point  $t$  on which we need to predict an enterprise attack of a particular event type (2) we use features from the darkweb forums  $\delta$  days prior to  $t$  and, (3) we use these features as input to a machine learning model to predict attack on  $t$ . So one of the main tasks involves learning the attack prediction model, one for each event type. We describe the attack prediction framework in the following section.

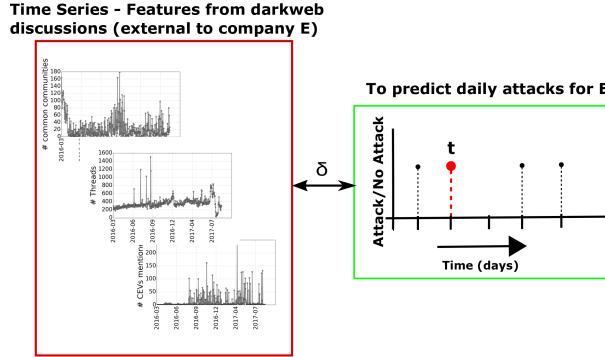


Fig. 5: The prediction task. We use unconventional time series signals from the darkweb forum network to predict attack on a daily basis for company  $E$ .

## 5 Framework for Attack Prediction

Since we attempt at building an integrated framework leveraging the network formed from the discussions in the forums as signals for predicting organization specific attacks, we segregate it into the three steps of any classic machine learning framework:

1. *Feature engineering*: As one of our contributions, we leverage the reply network formed from the thread replies in forums to build features for input to the model. To this end we build two kinds of features:
  - *Graph Based Features*: Here we identify features pertaining to the dynamics of replies from users with credible knowledge to regular posts
    - the intuition behind this is to see whether a post gaining attention from active and reputed users can be a predictive signal.
  - *Forum metadata*: We also gather some forum metadata as another set of features and we use them as baselines for our graph based features.
 So as a first step towards achieving this, we devise an algorithm to create the reply network structure from the replies in the threads in this step prior to feature computation.
2. *Training (learning) models for prediction*: In this step, we first split the timeframe of our attack study into two segments: one corresponding to the training span and the other being the test span. However, unlike normal cross-validated machine learning models, we need to be careful about the time split, since we consider longitudinal networks for features and the training-test split should respect the forecasting aspect of our prediction
  - we use features  $\delta$  days prior to the day we predict the attacks for. So instead of using cross-validation, we fix our training time span as the first few time points in our ground truth dataset (chronologically ordered) and the test span succeeding the training span. We build several time-series of individual features from step 1 using only forum discussions in the training

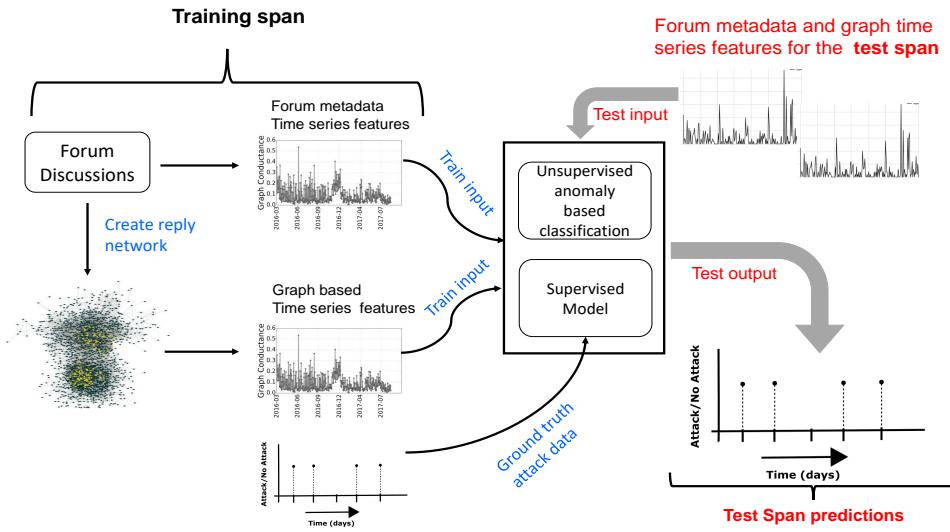


Fig. 6: An overview of the framework used for attack prediction.

span and use them as input along with the attack ground truth to a supervised model for learning the parameters (we build separate models for separate attack types and different attack organizations). This along with step 1 is shown in Figure 6 on the left side under the training span stage.

3. *Attack prediction:* In this final step, we first compute the time series of the same set of features in the test span, instead that we now use the forum discussions in the test span ( $\delta$  days prior to the prediction time point). We input these time series into the supervised model as well as an additional unsupervised model (that does not require any training using ground truth), to output attacks on a daily basis in the test span. This step is displayed in the right component of Figure 6.

In the following sections, we explain the steps in details that also describes the intuition behind the approach used for attack prediction in our study.

### 5.1 Step 1: Feature Engineering

For the purposes of network analysis, we assume the absence of global user IDs across forums<sup>8</sup> and therefore analyze the social interactions using networks induced on specific forums instead of considering the global network of all users across all forums. We denote the directed and unweighted reply graph of a forum  $f \in F$  by  $G^f = (V^f, E^f)$  where  $V^f$  denotes the set of users who

<sup>8</sup> Note that even in the presence of global user IDs across forums, a lot of anonymous or malicious users would create multiple profiles across forums and create multiple posts with different profiles, identifying and merging which is an active area of research.

posted or replied in some thread in forum  $f$  at some time in our considered time frame of data and  $E^f$  denotes the set of 3-tuple  $(u_1, u_2, rt)$  directed edges where  $u_1, u_2 \in V^f$  and  $rt$  denotes the time at which  $u_1$  replied to a post of  $u_2$  in some thread in  $f$ ,  $u_1 \rightarrow u_2$  denoting the edge direction. We emphasize that this notation of the network discards links between users of 2 different forums as we did not connect or merge threads posted in two separate forums. We denote by  $G_\tau^f = (V_\tau^f, E_\tau^f)$ , a temporal subgraph of  $G^f$ ,  $\tau$  being a time window such that  $V_\tau^f$  denotes the set of individuals who posted in  $f$  in that window and  $E_\tau^f$  denotes the set of tuples  $(v_1, v_2, rt)$  such that  $rt \in \tau$ ,  $v_1, v_2 \in V_\tau^f$ .

### 5.1.1 Constructing the reply network

We adopt an incremental analysis approach by splitting the entire set of time points in our frame of study (both for the training and test span) into a sequence of time windows  $\Gamma = \{\tau_1, \tau_2, \dots, \tau_Q\}$ , where each subsequence  $\tau_i$ ,  $i \in [1, Q]$  is equal in time span and the subsequences are ordered by their starting time points for their respective span. This streaming aspect of the reply networks and the feature computation is based on our observation that the significance of users (in terms of *important* posts in the forums) change very rapidly and for a one year span, computing features for a month based on historical information of users long time back is not convenient. From that perspective, we create evolving networks on a daily basis (but which incorporate historical knowledge), and compute features on a daily basis. However, in more realistic settings, the temporal resolution of these snapshots can be managed dynamically based on how often consecutive networks change significantly in terms of some distance metric as has been done in [32].

Next we describe the operations: *Create* - that takes a set of forum posts in  $f$  within a time window  $\tau$  as input and creates a temporal subgraph  $G_\tau^f$  and *Merge* - that takes two temporal graphs as input and merges them to form an auxiliary graph that incorporates historical information. To keep the notations simple, we would drop the symbol  $f$  when we describe the operations for a specific forum in  $F$  as context but which would apply for any forum  $f \in F$ . We describe the two operations that describe how we map the features extracted from network structure  $G^f$  to a time series, the analysis of which is the one of contributions of our research:

1. *CREATE*: In this step, we create the reply network based on individual threads within a forum  $f$  on a daily basis. Let  $h$  be a particular thread or topic within a forum  $f$  containing posts by users  $V_h^f = \{u_1, \dots, u_k\}$  posted at corresponding times  $T_h^f = \{t_1, \dots, t_k\}$ , where  $k$  denotes the number of posts in that thread and  $t_i \geq t_j$  for any  $i > j$ , that is the posts are chronologically ordered. Since we are considering a reply network on the forum posts, the lack of information as to who replied to whom necessitates the use of some heuristics to connect the users based on temporal and spatial information. We note that in situations where the data comes with the hierarchical reply structure of who-replies-to-whom, this step can be

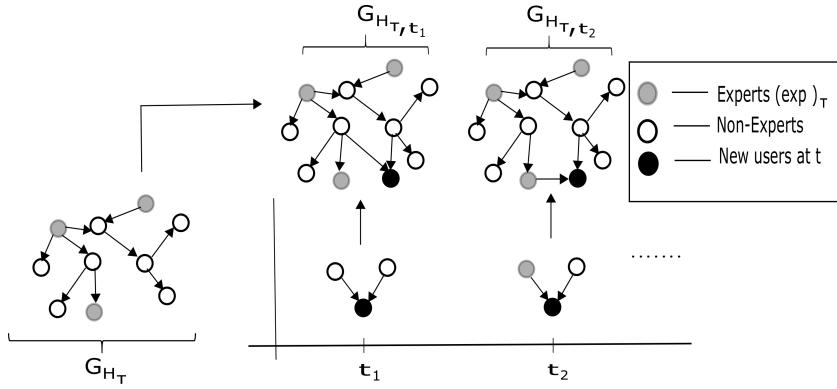


Fig. 7: An illustration to show the *Merge* operation:  $G_{H_\tau}$  denotes the historical network using which the experts shown in gray are computed.  $\{G_{t_1}, G_{t_2}, \dots\}$  denote the networks at time  $t_1, t_2, \dots \in \tau, \tau \in \Gamma$ . Note that the experts are extracted only from  $G_{H_\tau}$  and not on a regular basis.

avoided and can be skipped to the next stage. A simple approach would be to consider either (i) a *temporal constraint*: for each user  $u_i$  of a post in a thread  $h$  in forum  $f$  at time  $t_i$ , we would create an edge  $(u_i, u_k, t)$  such that  $t_i - t_k < \text{thresh}_{\text{temp}}$ ,  $u_k$  denotes the user for the respective posts at time  $t_k \in \tau$ ,  $\text{thresh}_{\text{spat}}$  denoting a time threshold or (ii) a *spatial constraint*: consider all edges  $(u_i, u_k, t_i)$ , where  $u_k$  denotes the user of the  $k^{\text{th}}$  post in the time ordered sequence of posts and  $k - i \leq \text{thresh}$ ,  $\text{thresh}$  denoting a count threshold. The idea behind reply edge construction based on the combination of these two constraints is the following: in a time interval where there are a lot of discussions, networks with the edges created from the condition bounded by  $\text{thresh}_{\text{temp}}$  would be unduly over-dense. Thus the second condition bounds the number of posts (prior to its current post) that a user can reach to while replying using its current post. In a way, this ensures normalization since the hypothesis here is that a user can only reach/reply to a certain number of posts prior to the current time irrespective of how popular the discussions might be in a specific time intervals.

We use both the constraints in the following way: for the  $i^{\text{th}}$  post  $p_{h,i}$  in the thread  $h$  posted at time  $t_i$ , the objective is to create links from the user of this post to the posts prior to this as reply links. For this, we consider a maximum of  $\text{thresh}_{\text{spat}}$  count of posts prior to  $p_{h,i}$  (note the posts in the thread are considered chronologically ordered), that is all posts  $p_{h,k}$  such that  $k - i \leq \text{thresh}_{\text{spat}}$ . The users for those respective posts would be the potential users to whom  $u_{h,i}$  replied to (unidirectional links), which we denote by  $\{u_{h,i \rightarrow k}\}$  and the corresponding set of posts  $\{p_{h,i \rightarrow k}\}$ . The next layer of constraints considering temporal boundaries prune out candidates from  $\{u_{h,k}\}$ , using the following two operations:

- If  $t_i - t_k < \text{thresh}_{\text{temp}}$ , we form edges linking  $u_{h,i}$  to all users in  $\{u_{h,i \rightarrow k}\}$  (note the direction of reply). This takes care of the first few posts in  $h$  where there might not be enough time to create a sensation, but anyhow the users might be replying as a general discussion in the thread. So we consider user of  $i^{\text{th}}$  post replies potentially to all these users of  $\{u_{h,i \rightarrow k}\}$  at one go whether it is at the beginning or whether it is in the middle of an ongoing thread discussion.
- If  $t_i - t_k \geq \text{thresh}_{\text{temp}}$ , we first compute the mean of the time differences between two successive posts in  $\{p_{h,i \rightarrow k}\}$ . We also denote the time difference between  $t_i$  and the time of the last post in  $\{p_{h,i \rightarrow k}\}$  considering the chronological ordering is maintained (this is the post prior to  $i$ ), as  $\Delta t_i$ . If the computed mean is less than  $\Delta t_i$ , we form edges linking  $u_{h,i}$  to all users in  $\{u_{h,i \rightarrow k}\}$  (this is similar to the first constraint). Else, as long as the mean is greater than  $\Delta t_i$ , we start removing the posts in  $\{p_{h,i \rightarrow k}\}$  farthest in time to  $t_i$  in order and recalculate the mean after removal of such posts. We repeat this procedure until at some iteration either the recomputed mean is less than  $\Delta t_i$  or  $t_i - t_k < \text{thresh}_{\text{temp}}$ . This heuristic considers the case for posts that receive a lot of replies very frequently at certain time of the thread lifecycle, although it is not reasonable to consider posts which have been posted a while ago as being replied to by the current post in consideration.

Following this,  $V^f = \cup_h V_h^f$  and  $E^f = \cup_h E_h^f$ , that is we remove multiple interactions between the same set of users in multiple threads and without weighting these edges. As before, a temporal subgraph of  $G^f$  would be denoted by  $G_\tau^f$  where  $(u, v, rt) \in E_\tau$  denotes  $u$  replied to  $v$  at time  $rt \in \tau$ . Our objective after creating the reply network  $G_\tau^f$  is to compute features from this network that could then be used as input to a machine learning model for predicting cyber attacks. These features would act as the unconventional signals that we have been addressing in this paper for predicting external enterprise specific attacks. In order to achieve that, we need to form time series of a feature  $x$  (among a set of network features) denoted by  $\mathcal{T}_{x,f}$  for every forum  $f \in F$  separately: formally  $\mathcal{T}_{x,f}$  is a stochastic process that maps each time point  $t$  to a real number.

2. **MERGE:** In order to create a time series feature  $\mathcal{T}_{x,f}$  for feature  $x$  from threads in forum  $f$ , we use 2 reply networks: (1) a historical network  $G_{H_\tau}$  which spans over time  $H_\tau$  such that  $\forall t' \in H_\tau$ , and for any  $t \in \tau$ , we have  $t' < t$ , and (2) the network  $G_t^f$  induced by user interactions between users in  $E_t$ , which varies temporally for each  $t \in \tau$ . We note that the historical network  $G_{H_\tau}$  would be different for each subsequence  $\tau$ , so as the subsequences  $\tau \in \Gamma$  progress with time, the historical network  $G_{H_\tau}$  also changes, and we discuss the choice of spans  $\tau \in \Gamma$  and  $H_\tau$  in Section 6. Finally, for computing feature values for each time point  $t \in \tau$ , we merge the 2 networks  $G_{H_\tau}$  and  $G_t^f$  to form the auxiliary network  $G_{H_\tau,t} = (V_{H_\tau,t}, E_{H_\tau,t})$ , where  $V_{H_\tau,t} = V_{H_\tau} \cup V_t$  and  $E_{H_\tau,t} = E_{H_\tau} \cup E_t$ . A visual illustration of this method is shown in Figure 7. At the end, we consider several network

features over each  $G_{H_\tau, t}^f$  and compute the feature values at time point  $t$  to form  $\mathcal{T}_{x, f}$  for feature  $x$  and forum  $f$ .

### 5.1.2 Network based features

We leverage the network  $G_{H_\tau}^f$  to compute features on a regular basis - the advantage is that this network contains historical information but at the same time, this historical information does not update on a regular basis. For extracting network based features, we want to be able to focus on the interactions convened by users in forums with a knack towards posting credible information. The objective is to investigate whether any spike in attention towards posts on a day from such users with some *credible reputation* translates to predictive signals for cyber attacks on an organization. This would also in a way help filter out noisy discussions or replies from unwanted or naive users who post information irrelevant to vulnerabilities or without any malicious intent. We hypothesize that predictive signals would exhibit users in these daily reply networks whose posts have received attention (in the form of direct or indirect replies) from some “expert” users - whether a faster reply would translate to an important signal for an attack is one of the novel questions we tackle here.

In order to be able to extract posts that receive attention on a daily basis, we first need to extract “expert” users who attention we seek to gather.

*Expert Users.* For each forum  $f$ , we use the historical network  $G_{H_\tau}^f$  to extract the set of *experts* relevant to timeframe  $\tau$ , that is  $exp_\tau^f \in V_{H_\tau}^f$ . First, we extract the top CPE groups  $CP_\tau^{top}$  in the time frame  $H_\tau$  based on the number of historical mentions of CVEs. These would be used as top CPEs for the span  $\tau$ . For this, we sort the CPE groups based on the sum of the CVE mentions that belong to the respective CPE groups and take the top 5 CPE groups by sum in each  $H_\tau$ . Using these notations, the experts  $exp_\tau^f$  from history  $H_\tau$  considered for time span  $\tau$  are defined as users in  $f$  with the following three constraints:

1. Users who have mentioned a CVE in their post in  $H_\tau$ . This ensures that the user engages in the forums with content that is relevant to vulnerabilities.
2. Let  $\theta(u)$  denote the set of CPE tags of the CVEs mentioned by user  $u$  in his/her posts in  $H_\tau$  and such that it follows the constraint: either  $\theta(u) \in CP_\tau^{top}$  where the user’s CVEs are grouped in less than 5 CPEs or,  $CP_\tau^{top} \in \theta(u)$  in cases where a user has posts with CVEs in the span  $H_\tau$ , grouped in more than 5 CPEs. This constraint filters out users who discuss vulnerabilities which are not among the top CPE groups in  $H_\tau$ .
3. The in-degree of the user  $u$  in  $G_{H_\tau}^f$  should cross a threshold. This constraint ensures that there are a significant number of users who potentially responded to this user thus establishing  $u$ ’s central position in the reply network. These techniques to filter out relevant candidates based on network topology has been widely used in the bot detection communities [36].

We avoid using other centrality metrics instead of using the in-degree in the third constraint since our focus here is not to judge the position of the user from the centrality perspective (for example, high betweenness would not denote the user receives multiple replies on its posts). Instead, we want to filter out users who receive multiple replies on their posts or in other words their posts receive attention. Essentially, these set of experts  $exp_\tau$  from  $H_\tau$  would be used for all the time points in  $\tau$  as shown in Figure 7. Our objective here is to not consider the degree as the proxy for user importance in any terms. Rather the degree indicates the number of replies it gets from other users.

*Why focus on experts?* To show the significance of these properties in comparison to other users, we perform the following hypothesis test: we collect the time periods of 3 widely known security events: the WannaCry ransomware attack that happened on May 12, 2017 and the vulnerability MS-17-010, the Petya cyber attack on 27 June, 2017 with the associated vulnerabilities CVE-2017-0144, CVE-2017-0145 and MS-17-010, the Equifax breach attack primarily on March 9, 2017 with vulnerability CVE-2017-5638. We consider two sets of users across all forums -  $exp_\tau$ , where  $G_{H_\tau}$  denotes the corresponding historical network prior to  $\tau$  in which these 3 events occurred and the second set of users being all  $U_{alt}$  who are not experts and who fail either one of the two constraints: they have mentioned CVEs in their posts which do not belong to  $CP^{top}$  or their in-degree in  $G_{H_\tau}$  lies below the threshold. We consider  $G_{H_\tau}$  being induced by users in the last 3 weeks prior to the occurrence week of each event for both the cases, and we consider the total number of interactions ignoring the direction of reply of these users with other users. Let  $\text{deg}_{\text{exp}}$  denote the vector of count of interactions in which the *experts* were involved and  $\text{deg}_{\text{alt}}$  denote the vector of counts of interactions in which the users in  $U_{alt}$  were involved. We randomly pick number of users from  $U_{alt}$  equal to the number of experts and sort the vectors by count. We conduct a 2 sample t-test on the vectors  $\text{deg}_{\text{exp}}$  and  $\text{deg}_{\text{alt}}$ . The null hypothesis  $H_0$  and the alternate hypothesis  $H_1$  are defined as follows;

$$H_0 : \text{deg}_{\text{exp}} \leq \text{deg}_{\text{alt}}, \quad H_1 : \text{deg}_{\text{exp}} > \text{deg}_{\text{alt}}$$

The null hypothesis is rejected at significance level  $\alpha = 0.01$  with  $p$ -value of 0.0007. This suggests that with high probability, experts tend to interact more prior to important real world cyber-security breaches than other users who randomly post CVEs.

Now, we conduct a second  $t$ -test where we randomly pick 4 weeks not in the weeks considered for the data breaches, to pick users  $U_{alt}$  with the same constraints. We use the same hypotheses as above and when we perform statistical tests for significance, we find that the null hypothesis is not rejected at  $\alpha=0.01$  with a  $p$ -value close to 0.05. This empirical evidence from the  $t$ -test also suggests that the interactions with  $exp_\tau$  are more correlated with an important cyber-security incident than the other users who post CVEs not in top CPE groups and therefore it is better to focus on users exhibiting our desired properties as experts for cyber attack prediction. Note that the

*t – test* evidence also incorporates a special temporal association since we collected events from three interleaved timeframes corresponding to the event dates and we did not select any timeframe to show the evidence.

Next, we describe the following graph based features that we use to compute  $\mathcal{T}_{x,f}[t]$  at time  $t$ , for which we also take as input the relevant experts  $exp_\tau$ . We describe 4 network features that capture this intuition behind the attention broadcast by these users - the idea is that a cyber-adversary looking to thwart the prediction models from working by curating similar reply networks using bots, would need to not only introduce such random networks but would also have to get the desired attention from these experts which could be far challenging to achieve given that human attention is known to be different compared to bots. [33].

*Graph Conductance.* As studied in [34, 35, 37], social networks are fast mixing; this means that a random walk on the social graph converges quickly to a node following the stationary distribution of the graph. Applied to social interactions in a reply network, the intuition behind computing the graph conductance is to understand the following: can we compute bounds of steps within which any attention on a post would be successfully broadcast from the non-experts to the *experts* when a post closely associated with an attack is discussed [38]? One way of formalizing the notion of **graph conductance**  $\phi$  is:  $\phi = \min_{X \subset V: \pi(X) < \frac{1}{2}} \phi_X$  where  $\phi_X$ ,  $X$  being the set of experts here is defined as:  $\phi_{Experts} = \frac{\sum_{x \in exp_\tau} \sum_{y \in V_t \setminus exp_\tau} \pi(exp_\tau) P_{xy}}{\pi(exp_\tau)}$ , and  $\pi(\cdot)$  is the stationary distribution of the network  $G_{H_\tau, t}$ . For subset of vertices  $exp_\tau$ , its conductance  $\phi_{Experts}$  represents the probability of taking a random walk from any of the *experts* to one of the users in  $V_t \setminus exp_\tau$ , normalized by the probability weight of being on an expert.

Applied to the reply network comprising both experts and the regular users, the key intuition behind conductance as used here is: *the mixing between expert nodes and the users of important posts is fast, while the mixing between expert nodes and regular nodes without important posts (in our view of importance as seeking attention) is slow*. So higher the value of conductance here, higher is the probability that the experts are paying attention to the posts and so there is a good chance that the conversations on those days could be reflective of a cyber attack in future.

*Shortest paths.* To understand the dynamics of distance between the non-experts and the set of experts prior to an attack, we compute the shortest distance metric between them as follows:  $SP(exp_\tau, V_t \setminus exp_\tau) = \frac{1}{|exp_\tau|} \sum_{e \in exp_\tau} \min_{u \in V_t \setminus exp_\tau} s_{e,u}$ , where  $s_{e,u}$  denotes the shortest path in the graph  $G_{H_\tau, d}$  from the expert  $e$  to a user  $u$  in the direction of the edges. Since the edges are formed in the direction of the replies based on time constraints, it also denotes how fast an expert replies in a thread that leads back in time to a post by  $u$ . Such distance metrics have been widely used in network analysis to understand the pattern of interactions [39].

**Algorithm 1:** Algorithm for computing Common Communities (CC)

---

```

Input:  $exp_{\tau}, G_{H_{\tau}}, (V_t, E_t)$ 
Output:  $CC(exp_{\tau}, V_t \setminus exp_{\tau})$  - the number of communities shared by  $V_t \setminus exp_{\tau}$ 
          with  $exp_{\tau}$  at  $t$ 

1 communities = Louvain_community( $G_{H_{\tau}}$ ) ;      // dictionary storing node to
   community index mapping
2  $c_{expSet} \leftarrow ()$  ;
3 foreach user  $u \in exp_{\tau}$  do
4   |  $c_{expSet}.add(communities[u])$  ;
5 end
6  $V_{H_{\tau},t} \leftarrow V_{H_{\tau}} \cup V_t$  ;
7  $E_{H_{\tau},t} \leftarrow E_{H_{\tau}} \cup E_t$  ;
8  $CC(exp_{\tau}, V_t \setminus exp_{\tau}) \leftarrow 0$  ;           // stores count
/* Iterate over the users in  $V_t$  who have not been assigned communities
   from  $H_{\tau}$  */
9 foreach user  $u \in V_t$  do
10  | if  $u \in V_{H_{\tau}}$  and  $communities(u) \in c_{expSet}$  then
11    |   |  $CC(exp_{\tau}, V_t \setminus exp_{\tau}) += 1$ ;
12  | end
13  | else
14    |   | foreach user  $v \in exp_{\tau}$  do
15      |     | /* Condition 1 */
16      |       | if  $(v, u) \in E_{H_{\tau},t}$  then
17      |         |   |  $CC(exp_{\tau}, V_t \setminus exp_{\tau}) += 1$ ;
18      |         |   | break ;
19      |       | end
20      |     | /* Condition 2 */
21      |       | foreach user  $n \in inNeighbors(E_{H_{\tau},t}, u)$  do
22      |         |   | if  $communities(n) \in c_{expSet}$  then
23      |           |   |   |  $CC(exp_{\tau}, V_t \setminus exp_{\tau}) += 1$ ;
24      |           |   |   | break ;
25      |       | end
26    |   | end
27 end
28 return  $CC(exp_{\tau}, V_t \setminus exp_{\tau})$ 

```

---

*Expert Replies.* To analyze whether *experts* reply to users more actively when there is an important discussion going on surrounding any vulnerabilities or exploits, we compute the number of replies by an *expert* to users in  $V_t \setminus exp_{\tau}$ . We calculate the number of out-neighbors of  $exp_{\tau}$  considering  $G_{H_{\tau},t}$ .

*Common Communities.* To evaluate the role of communities in the reply network and to assess whether experts engage with selected other users within a community when an information gains attention and could be related to vulnerability exploitation, we use community detection on the networks  $G_{H_{\tau}}$ . We use the Louvain method [40] to extract the communities from a given network. Since it is not computationally feasible to compute communities in  $G_{H_{\tau},t}$  for all the time points  $t \in \tau$ , we first compute all the communities for the users in the historical network  $G_{H_{\tau}}$ . Following this, we use an approximation based

Group	Features	Description
Expert centric	Graph Conductance	$\tau_x[t] = \frac{\sum_{x \in \exp_\tau} \sum_{y \in V_t \setminus \exp_\tau} \pi(\exp_\tau) P_{xy}}{\pi(\exp_\tau)}$ where $\pi(\cdot)$ is the stationary distribution of the network $G_{H_\tau, t}$ , $P_{xy}$ denotes the probability of random walk from vertices $x$ to $y$ . The conductance represents the probability of taking a random walk from any of the experts to one of the users in $V_t \setminus \exp_\tau$ , normalized by the probability weight of being on an expert.
	Shortest Path	$\tau_x[t] = \frac{1}{ \exp_\tau } \sum_{e \in \exp_\tau} \min_{u \in V_t \setminus \exp_\tau} s_{e,u}$ where $s_{e,u}$ denotes the shortest path from an expert $e$ to user $u$ following the direction of edges.
	Expert replies	$\tau_x[t] = \frac{1}{ \exp_\tau } \sum_{e \in \exp_\tau}  \text{OutNeighbors}(e) $ where $\text{OutNeighbors}(\cdot)$ denotes the out neighbors of user in the network $G_{H_\tau, t}$ .
	Common Communities	$\tau_x[t] =  \mathcal{N}(c(u)) \mid c(u) \in c_{\text{experts}} \wedge u \in V_t \setminus \exp_\tau\}$ where $c(u)$ denotes the community index of user $u$ , $c_{\text{experts}}$ that of the experts and $\mathcal{N}(\cdot)$ denotes a counting function. It counts the number of users who share communities with experts.
Forum/User Metadata	Number of threads	$\tau_x[t] =  \{h \mid \text{thread } h \text{ was posted on } t\} $
	Number of users	$\tau_x[t] =  \{u \mid \text{user } u \text{ posted on } t\} $
	Number of expert threads	$\tau_x[t] =  \{h \mid \text{thread } h \text{ was posted on } t \text{ by users } u \in \exp_\tau\} $
	Number of CVE mentions	$\tau_x[t] =  \{CVE \mid \text{CVE was mentioned in some post on } t\} $

Table 2: List of features used for learning. Each feature  $\tau_x$  is computed separately across forums.

on heuristics to compute the communities of new users  $V_{new} = V_{H_\tau, t} \setminus V_{H_\tau}$ . Let  $c_{\text{experts}}$  denote the set of communities that users in  $\exp_\tau$  belong to following the call to Louvain method in Line 1 of Algorithm 1. Let  $c(u)$  denote the community index of a user  $u$ . We define the common communities measure as follows:  $CC(\exp_\tau, V_t \setminus \exp_\tau) = \{N(c(u)) \mid c(u) \in c_{\text{experts}} \wedge u \in V_t \setminus \exp_\tau\}$ , that is it measures the number of non-experts at time  $t \in \tau$  that share the same communities with  $\exp_\tau$ . We use 2 approximation constraints demonstrated in Lines 16-25 of Algorithm 1 to assign a new user  $u \in V_{new}$  to an expert community as follows:

1. *Condition 1*: If an expert has an incoming edge to  $u$ , we increase the count of common communities by 1.
2. *Condition 2*: If  $u$  has a incoming neighbor who shares a community in the set of communities of experts, we increase the count of common communities by 1. This is shown in Line 19 in the call to the *InNeighbors()* method.

### 5.1.3 User/Forum Metadata features

In addition to the network features, we compute the following forum based statistics for a forum  $f$  at time point  $t$ : (1) The number of unique vulnerabilities mentioned in  $f$  at time  $t$ , (2) The number of users who posted in  $f$ , (3)

the number of unique threads in  $f$  at time  $t$ , and (4) The number of threads in which there was at least one *expert* post among all the posts in  $f$  at  $t$ .

A brief summary of all the features used in this study is shown in Table 2.

## 5.2 Training models for prediction

In this section, we explain how we use the time series features  $\mathcal{T}_{x,f}$  across forums in  $F$  described in the preceding section to predict an attack at any given time point  $t$ . We consider 2 models for our framework: (1) a supervised learning model in which the time series  $\mathcal{T}_x$  is formed by averaging  $\mathcal{T}_{x,f}$  across all forums in  $f \in F$  at each time point  $t$  and then using machine learning models for the prediction task and, (2) an unsupervised learning model in which we take the time series  $\mathcal{T}_{x,f}$  for each feature and each forum  $f$  separately and then use dimensionality reduction techniques across the forums dimension. Following this, we use anomaly detection methods for the prediction task - this model does not use the training span ground truth attack data and directly works on features in the training and test span to predict attacks. However, in the supervised learning scenario we build separate prediction models for each attack type in  $A$  and for each organization separately. We do not use the two learning models in conjunction nor do we combine data from different attack types together - we leave that as a future work to see how models built on one attack type could generalize to other types and whether we can use different attack types together as a multi-label classification problem although such models of synthesis have been used previously for attack prediction[41]. We treat the attack prediction problem in this paper as a binary classification problem in which the objective is to predict whether there would be an attack at a given time point  $t$  (Refer Figure 5). Since the incident data in this paper contains the number of incidents that occurred at time point  $t$ , we assign a label of 1 for  $t$  if there was at least one attack at  $t$  and 0 otherwise.

### 5.2.1 Supervised Learning

We first discuss the technical details of the machine learning model that learns parameters based on the given training labels of different attack types in  $A$  in the training span and uses them to predict whether an organization  $E$  would be vulnerable to an attack of some type in  $A$  at  $t$  - we note again that we build different models for each attack type in  $A$  for  $E$ , so predicting for each type means that we have to learn different models for the types, however the set of time series features gathered in the previous step as input is consistent across all models. In [53, 43], the authors studied the effect of longitudinal sparsity in high dimensional time series data, where they propose an approach to assign weights to the same features at different time spans to capture the temporal redundancy. We use 2 parameters:  $\delta$  that denotes the start time prior to  $t$  from where we consider the features for prediction and  $\eta$ , the time span (window)

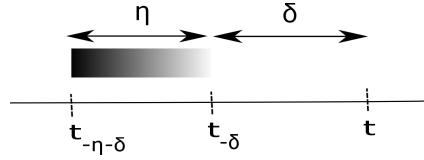


Fig. 8: Temporal feature selection window for predicting an attack at time  $t$

for the features to be considered. An illustration is shown in Figure 8 where to predict an attack occurrence at time  $t$ , we use the features for each time  $t \in [t_{-\eta-\delta}, t_{-\delta}]$ . We use logistic regression with longitudinal ridge sparsity that models the probability of an attack as follows:

$$P(\text{attack}(t) = 1 | \mathbf{X}) = \frac{1}{1 + e^{-(\beta_0 + \sum_{k=\eta+\delta}^{\delta} \beta_k x_{t-k})}} \quad (1)$$

The final objective function to minimize over  $N$  instances where  $N$  here is the number of time points spanning the attack time frame is :  $l(\beta) = -\sum_{i=1}^N (y_i(\beta_0 + \mathbf{x}_i^T \beta) - \log(1 + \exp^{\beta_0 + \mathbf{x}_i^T \beta}) + \lambda \beta^T \beta. \mathbf{T}$

To obtain the aggregate series  $\mathcal{T}_x$  from individual forum features  $\mathcal{T}_{x,f}$ , we just average the values across all forums for each time point. Here we use each feature separately although later we discuss the combinations of features together with sparsity constraints in Section 6.2.3.

### 5.2.2 Unsupervised Learning

Now, we discuss the unsupervised learning model that directly takes as input the time series features in the training span as input and predicts the attacks for types in  $A$  on an organization  $E$  in the test span. However, unlike the supervised model, this model's prediction output does not depend on the type of attacks or the organization -  $E$ . It produces the same output for any attack - we try to see how do anomalies from such unconventional signals in the darkweb correlate with the attacks in the real world. Informally, anomalies are patterns in data that do not conform to a well defined notion of normal behavior. The problem of finding these patterns is referred to as anomaly detection [44, 45]. The importance of anomaly detection comes from the idea that anomalies in data translate to information that can explain actionable deviations from normal behavior thus leading to a cyber attack. We use subspace based anomaly detection methods that take as input,  $\mathcal{T}_{x,f}$ , aggregates them across all forums and finds anomalies in the cumulative time series for feature  $x$ . We derive motivation for this technique from the widely used projection based anomaly detection methods [46, 47] that detects volume anomalies from the time series of network link traffic. Additionally, there have been techniques in graph based anomaly detection that finds graph objects that are rare and considered outliers [48]. However, our motivation behind using anomaly detection does not lie from a feature analysis perspective or finding anomalous users

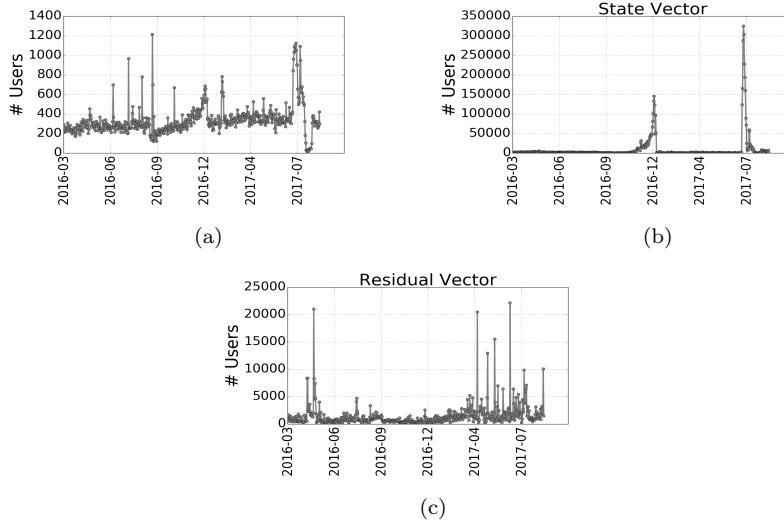


Fig. 9: (a) The time series  $\mathcal{T}$  for the number of users feature computed on a daily basis and averaged across all forums  $F$  (b) The SPE *state* time series vector after subspace separation (not averaged) (c) The SPE residual time series vector  $\mathcal{R}$  after subspace separation (not averaged).

but from a time series perspective - we observed that there could be spikes in time series of the same feature in different forums on different days. The question is how do we aggregate information from these spikes together instead of averaging them to an extent that the spikes die out in the aggregate. From that perspective, we find that the method used in [46] suits our framework - we want to be able to filter out the spikes from the same feature computed in different forums while projecting the dimension space of several forums to a 1-dimensional subspace. The overall procedure for detecting anomalies from the time series data on each feature has been described through the following steps. We will again drop the subscript  $x$  to generalize the operations for all features.

*Aggregating time series.* We create a matrix  $\mathbf{Y}$  with dimensions (# time points)  $\times$  ( $F$ ), the rows denoting values at a single time step  $t$  for forums  $f \in F$ . While  $\mathbf{Y}$  denotes the set of measurements for all forums  $F$ , we would also frequently work with  $\mathbf{y}$ , a vector of measurements from a single timestep  $t$ .

*Subspace Separation.* Principal Component Analysis (PCA) [49] is a method to transform the coordinates of the data points by projecting them to a set of new axes which are termed as the principal components. We apply PCA on matrix  $\mathbf{Y}$ , treating each row of  $\mathbf{Y}$  as a point in  $\mathbb{R}^F$ . Applying PCA to  $\mathbf{Y}$

yields a set of  $F$  principal components,  $\{\mathbf{v}_i\}_{i=1}^F$ . In general, the  $k$ th principal component  $\mathbf{v}_k$  is:  $\mathbf{v}_k = \arg \max_{\|\mathbf{v}\|=1} \|(\mathbf{Y} = -\sum_{i=1}^{k-1} \mathbf{Y}\mathbf{v}_i\mathbf{v}_i^T)v\|$ . We determine the *principal axes (components)* by choosing the first few components that capture the maximum variance along their direction. Once these *principal axes* have been determined, the matrix  $\mathbf{Y}$  can be mapped onto the new axes leading to as *residual or anomalous subspace*.

For detecting anomalies, we need to separate the vectors  $\mathbf{y} \in \mathbb{R}^F$  at any timestep into normal and anomalous components. We will refer to these as the *state* and *residual* vectors of  $\mathbf{y}$ . The key idea in the subspace-based detection step is that, once  $\mathcal{S}$  and  $\tilde{\mathcal{S}}$  have been constructed, the separation can be done by projecting  $\mathbf{y}$  onto these subspaces. We tend to decompose this  $\mathbf{y}$  as:  $\mathbf{y} = \hat{\mathbf{y}} + \tilde{\mathbf{y}}$ . For this, we arrange the set of principal components corresponding to the normal subspace ( $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r$ ) as columns of a matrix  $\mathbf{P}$  of size  $f \times r$  where  $r$  denotes the number of *normal principal axes* determined from the previous step. We can then form  $\hat{\mathbf{y}}$  and  $\tilde{\mathbf{y}}$  as:

$$\hat{\mathbf{y}} = \mathbf{P}\mathbf{P}^T\mathbf{y} = \mathbf{C}\mathbf{y} \quad \text{and} \quad \tilde{\mathbf{y}} = (\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{y} = \tilde{\mathbf{C}}\mathbf{y} \quad (2)$$

where the matrix  $\mathbf{C} = \mathbf{P}\mathbf{P}^T$  represents the linear operator that performs projection onto the normal subspace, and  $\tilde{\mathbf{C}}$  likewise projects onto the residual subspace. Here  $\hat{\mathbf{y}}$  is referred to as the state vector and  $\tilde{\mathbf{y}}$  as the residual vector.

*Detection of anomalies.* The idea of anomaly detection is to monitor the residual vector that captures abnormal changes in  $\mathbf{y}$ . As mentioned in [46, 50], there have been substantial research into designing statistical metrics for detecting abnormal changes in  $\tilde{\mathbf{y}}$  using thresholding and we use one of the widely used metrics, the squared prediction error (SPE) on the residual vector:  $SPE \equiv \|\tilde{\mathbf{y}}\| \equiv \|\tilde{\mathbf{C}}\mathbf{y}\|^2$ . This gives the SPE residual vector and when combined over all time points gives us the residual vector time series denoted by  $\mathcal{R}$ . The SPE residual vector at any time point is considered normal if  $SPE \leq \delta_\alpha^2$ , where  $\delta_\alpha^2$  denotes the threshold for the SPE at the  $1 - \alpha$  confidence level. We keep this threshold dynamic and would use it as a parameter for evaluating the anomaly based prediction models later on described in Section 6. Figure 9 demonstrates the decomposition of the time series into the SPE state and residual vectors. While Figure 9(b) captures most of the normal behavior, the SPE residual time series in Figure 9(c) captures all the anomalies across all the forums. The key point of this anomaly detection procedure is that instead of monitoring the time series feature  $\mathcal{T}_{x,f}$  separately across all forums in  $F$  for predicting cyber attacks, we have reduced it to monitoring the SPE residual time series  $\mathcal{R}_x$  for cyber attacks.

### 5.3 Attack prediction

*Anomaly detection to Attack prediction.* Following the subspace projection method to obtain  $\mathcal{R}_x$  denoting the SPE residual vector, from the input time

series feature  $\mathcal{T}_{x,f}$  for all forums  $f \in F$ , we use threshold mechanisms on  $\mathcal{R}_x$  to flag the time point  $t$  as an anomaly if  $\mathcal{R}_x[t]$  is greater than a threshold value. Given any test time point  $t$  as the test instance, we first project the times series vector  $\mathcal{T}_x[t_{-(\eta+\delta)} : t_{(-\delta)}]$  that contains the information of feature  $x$  across all forums in  $F$ , on the anomalous subspace  $\tilde{\mathbf{C}} = \mathbf{I} - \mathbf{P}\mathbf{P}^T$  given in Equation 2, if that time window is not already part of the training data. Following this, we calculate the squared prediction error (SPE) that produces a 1-dimensional vector  $\mathbf{y}_{test}$  of dimension  $\mathbb{R}^{n \times 1}$ . We count the number of anomalous time points  $t_a$ , denoted by  $\mathcal{N}(t_a)$ , with  $t_a \in [t_{-(\eta+\delta)}, t_{(-\delta)}]$ , time points that cross a chosen threshold. Finally, we flag an attack at  $t$  if  $\mathcal{N}(t_a) \geq \max(1, \frac{\zeta}{7})$ . This metric gives a normalized count threshold over a week for any  $\zeta$  and for this window parameter  $\zeta$  being less than a week, we just count whether there is at least one anomaly in that time gap. The fact that we avoid the attack ground truth data to learn event based parameters has some pros and cons : while in the absence of sufficient data for training supervised models, such anomaly detectors can serve a purpose by investigating various markers or features for abnormal behavior leading to attack, the disadvantage is such methods cannot be tailored to specific events or specific attack types in organizations.

*Supervised model prediction.* For the logistic regression model, we first create the features time series  $\mathcal{T}_x$  for the test span and use it to calculate the probability of attack in Equation 1. When the probability is greater than 1, we output a positive attack case else we predict a no-attack case.

## 6 Experiments and Results

### 6.1 Parameter settings

In our work, the granularity for each time index in the  $\mathcal{T}$  function is 1 day, that is we compute feature values over all days in the time frame of our study. For incrementally computing the values of the time series, we consider the time span of each subsequence  $\tau \in \Gamma$  as 1 month, and for each  $\tau$ , we consider  $H_\tau = 3$  months immediately preceding  $\tau$ . That is, for every additional month of training or test data that is provided to the model, we use the preceding 3 months to create the historical network and compute the corresponding features on all days in  $\tau$ . As mentioned earlier, this streaming nature of feature generation ensures we engineer the features relevant to the timeframe of attack prediction. For choosing the experts with an in-degree threshold, we select a threshold of 10 (we tried the values in the list [5, 10, 15, 20]) to filter out users having in-degree less than 10 in  $G_{H_\tau}$  from  $\exp_\tau$ . We obtain this threshold by manually investigating a few experts in terms of their content of posts and we find that beyond a threshold of 10, a lot of users get included whose posts are not relevant to any malicious information or signals.

For the reply network construction, we have 2 parameters:  $thresh_{spat}$  and  $thresh_{temp}$  corresponding to the spatial and temporal constraints. For setting

both these constraints, we used a 2D grid search over these parameters by constructing the reply network using pairwise combinations of these 2 parameters. Following this, for each combination we fit the in-degree distribution to power law with an exponent of 1.35. We fix the power law exponent based on a study [51] done where the authors found that a reply network which was created when the thread reply hierarchy was known in 2 forums, was best fit to a power law (in-degree distribution) when the exponents were in the range [1.35, 1.75]. We take the pair combination which gives us the minimum difference when we calculate the error arising from our degree distribution and  $p(k) \sim k^{-1.35}$ . Using this procedure we found  $\text{thresh}_{\text{spat}}=10$  (posts) and  $\text{thresh}_{\text{temp}}=15$  (minutes) to have the best fit in terms of the reply network we created.

The hyper-parameters for the logistic regression model  $\eta$  and  $\delta$  have been selected using a cross validation approach which we discuss briefly in the Results section. Similarly for detection of anomalies, the threshold parameter for the residual vector  $\delta_\alpha^2$  mentioned in Section 5.3, we test it on different values and plot the ROC curve to test the performance. For the choice anomaly count threshold parameter  $\zeta$ , such that we tag a cyber attack on  $t$  when the count of anomalies in the selected window  $t_{-\eta-\delta}, t_{-\delta}$  crosses  $\zeta$ , we set it to 1. The reason behind this is from manual observation where we find very days on which there are spikes and therefore, as a simple method, we just attribute an attack to a day if there was at least one anomaly in the time window prior to it. We do realize that this parameter needs to be cross-validated but our observations suggest that there would be very low precision in the performance when  $\zeta$  is set to a high value.

## 6.2 Results

To demonstrate the effectiveness of the features on real world cyber attacks, we perform separate experiments with the learning models described in Section 5.2: while for the anomaly detection based prediction, we use the same set of features as the only input for attack prediction across different attack types, for the supervised model, we build different learning models using the ground truth available from separate attack types in  $A$ . Additionally we only perform supervised classification for the *malicious-email* and the *endpoint-malware* attack types leaving out *malicious-destination* due to lack of sufficient training data. As mentioned in Section 5.2, we consider a binary prediction problem in this paper - we assign an attack flag of 1 for at least 1 attack on each day and 0 otherwise have the following statistics: for *malicious-email*, out of 335 days considered in the dataset, there have been reported attacks on 97 days which constitutes a positive class ratio of around 29%, for *endpoint-malware* the total number of attack days are 31 out of 306 days of considered span in the training dataset which constitutes a positive class ratio of around 10%, and for *endpoint-malware* we have a total of 26 days of attack out of a total of 276 days considered in the training set that spanned those attack days consti-

	Train positive sample	Train negative samples	Test positive samples	Test negative samples
Malicious email	65	178	32	60
Endpoint Malware	49	134	31	92
Malware Destination	7	115	8	84

Table 3: Statistics of the training and test samples from Armstrong.

tuting a positive class ratio of 9.4%. Table 3 shows the statistics of the training and test data for the 3 cyber attacks types from Armstrong. Although we did not use remedial diagnostics in our learning models to account for this class imbalance, the absence of a large training dataset and the missing attack data information accounting for irregularities make a strong case for using sampling techniques to address these issues which we leave as a future research direction for cyber attack focused studies. One of the challenges in remedial diagnostics for imbalances in classes is that here we need to take into account the temporal dependencies while incorporating any sampling techniques as remedies. However, we run a complementary experiment using SMOTE sampling as a simple measure for introducing synthetic samples into the training dataset which we discuss in Section 6.2.2.

For evaluating the performance of the models on the dataset, we split the time frame of each event into 70%-30% averaged to the nearest month separately for each *event-type*. That is we take the first 70% of time in months as the training dataset and the rest 30% in sequence for the test dataset. We avoid shuffle split as generally being done in cross-validation techniques in order to consider the consistency in using sequential information when computing the features. As shown in Figures 1, since the period of attack information provided varies in time for each of the events, we use different time frames for the training model and the test sets. For the event *malicious email* which remains our primary testbed evaluation event, we consider the time period from October 2016 to May 2017 (8 months) in the darkweb forums for our training data and the period from June 2017 to August 2017 (3 months) as our test dataset, for the *endpoint-malware*, we use the time period from April 2016 to September 2016 (6 months) as our training time period and June 2017 to August 2017 (3 months) as our test data for evaluation.

### 6.2.1 Unsupervised model prediction performance

Here we use the subspace projection method described in Sections 5.2.2, to filter out anomalies from the SPE residual time series vector  $\mathcal{R}_x$ . We then use these anomalies to predict the attacks as described there and try to see the tradeoffs between the number of true alerts and the number of false alerts obtained. We consider the first 8 principal components among the 53 forums that we considered. Among them we used the first 3 as the *normal axes* and

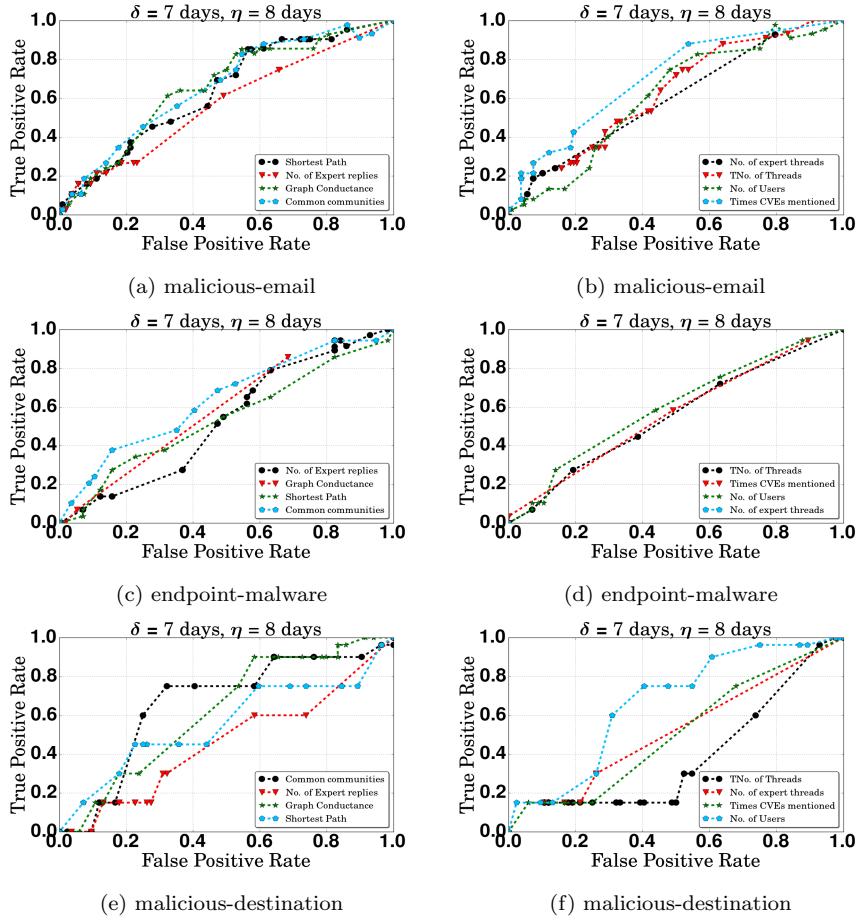


Fig. 10: ROC curves for prediction using unsupervised anomaly detection methods:  $\delta = 7$  days,  $\eta=8$  days

the rest 5 as our *residual axes* based on empirical evidence that shows these 3 components capture the maximum variance.

For evaluating the prediction performance, we examined the ROC (Receiver Operating Characteristic) curves for the features over different spans of  $\delta$  and  $\eta$  but we present our keys findings from the case where we set  $\eta=8$  days and  $\delta=7$  days shown in Figure 10 although we did not find general conclusions over the choices of the parameters  $\eta$  and  $\delta$  from the results. Each point in these ROC curves denotes a threshold among a set of values chosen for flagging a point in the vector obtained from the squared prediction error of the projected test input  $\mathbf{y}$ , that crosses the threshold as an anomaly. We present the results in each plot grouped by the *event – type* and the feature classes: forum statistics and graph based statistics. From the Figures 10(a) and (b),

for the event type *malicious – email*, we obtain the best *AUC (Area Under Curve)* results of 0.67 for the vulnerability mentions by users feature among the forum statistics groups and an AUC of 0.69 for graph conductance among the set of graph based features. For the event type *malicious – destination*, we obtained a best AUC of 0.69 for the common community count feature among the set of graph based features and a best AUC of 0.66 on the number of users at  $t_d$  among the forum statistics. For the event-type *endpoint – malware*, we obtain a best AUC of 0.69 on the number of users stats and 0.63 on the common communities *CC* feature. Empirically, we find that among the network features examined that rely on the set of *experts*, it is not sufficient to just look at how these experts reply to other users in terms of frequency, shown by the results where they exhibit the least AUC in the unsupervised setting that we considered. The fact that common communities and the graph conductance turn out to be better predictors than just the shortest path distance or the number of replies by experts, suggest that *experts* tend to focus on posts of a few individuals when any significant post arises and hence, focusing on individuals who are close to these users in terms of random walks and communities would be favorable.

One of the reasons behind the poor performance of the detector on the *malicious – destination* type of attacks compared to *malicious – email* although the total number of incidents reported for both of them are nearly the same is that the average number of incidents for any week of attack for the 3 attack-types are: for *malicious – email*, we have an average of 2.9 attacks per week, for *endpoint – malware*, we have an average of 3.6 attacks per week and for *malicious – destination*, there are an average of 1.52 attacks per weeks. So although the number of incidents are similar, the number of days of attacks on which the attack occurs is lesser for *malicious – destination* attacks and which is important for the binary classification problem considered here.

### 6.2.2 Supervised model prediction performance

For the logistic regression model, we consider a span of 1 week time window  $\eta$  while keeping  $\delta = 8$  days similar to the unsupervised setting. Due to absence of sufficient positive examples, we avoid using this model for predicting attacks of type *malicious – destination*. From among the set of statistics features that were used for predicting *malicious – email* attacks shown in Figure 11(b), we observe the best results using the number of threads as the signal for which we observe a precision of 0.43, recall of 0.59 and an F1 score of 0.5 against the random F1 of 0.34 for this type of attacks. From among the set of graph based features, we obtain the best results from graph conductance with a precision of 0.44, recall of 0.65 and an F1 score of 0.53 which shows an increase in recall over the number of threads measure. Additionally, we observe that in case of supervised prediction, the best features in terms of F1 score are graph conductance and shortest paths whereas number of threads and vulnerability mentions turn out to be the best among the statistics. For the attacks belonging to the type *endpoint – malware*, we observe similar characteristics

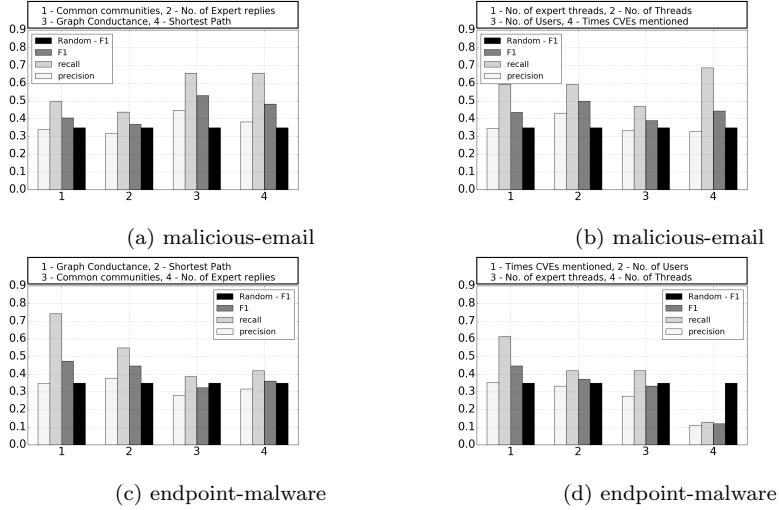


Fig. 11: Classification results for the features considering the supervised model:  $\delta = 7$  days,  $\eta = 8$  days.

for the graph features where we obtain a best precision of 0.34, recall of 0.74 and an F1 score of 0.47 against a random F1 of 0.35, followed by the shortest paths measure. However for the statistics measures we obtain a precision of 0.35, recall 0.61 and an F1 score of 0.45 for the vulnerability mentions followed by the number of threads which gives us an F1 score of 0.43. Although the common communities features doesn't help much in the overall prediction results, in the following section we describe a special case that demonstrates the predictive power of the community structure in networks. The challenging nature of the supervised prediction problem is not just due to the issue of class imbalance, but also the lack of large samples in the dataset which if present, could have been used for sampling purposes. As an experiment, we also used Random Forests as the classification model, but we did not observe any significant improvements in the results over the random case, suggesting the LR model with temporal regularization helps in these cases of time series predictions.

Additionally, we use SMOTE to deal with the class imbalance and we plot the results for the malicious email attacks in Figure 12 - from the results and comparing them with those in Figure 11, we find that while for all features the recall increases, the precision drops substantially. We find that among the graph features, both graph conductance and the number of expert replies perform equally well with an F1 score of 0.52 while the number of threads with CVE mentions achieves the best results with an F1 score of 0.49.

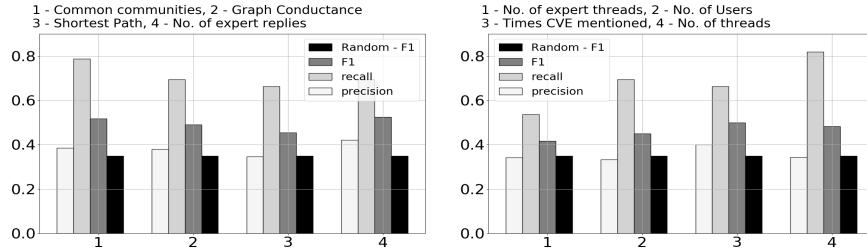


Fig. 12: Classification results for the malicious-email attack dataset using SMOTE sampling on top of the supervised learning model

### 6.2.3 Model with feature combinations

One of the major problems of the dataset is the imbalance in the training and test dataset as will be described in Section 6. The added complexities arise from the fact that if we consider all features over the time window of feature selection, then the total number of features  $z$  (variables) for the learning models is:  $z = \# \text{ features} \times (\eta)$ . In our scenario, this would typically be almost equal to the number of data points we have for training depending on  $\eta$  and also depending on whether we consider different variations of the features in Table 2, which might result in overfitting. So in order to use all features in each group together for prediction, we use 3 additional regularization terms in the longitudinal regression model : the L1 penalty, the L2 penalty and the *Group Lasso* regularization [52]. We adapt this framework of regularization to our set of features following previous studies on lasso for longitudinal data [53] and the final objective function can be written as:

$$l(\beta) = - \sum_{i=1}^N \log(1 + e^{-y_i(\beta^T \mathbf{x}_i)}) + \frac{m}{2} \|\beta\|_2^2 + l \|\beta\|_1 + g.GL(\beta) \quad (3)$$

where  $m$ ,  $l$  and  $g$  are the hyper-parameters for the regularization terms and the  $GL(\beta)$  term is  $\sum_{g=1}^G \|\beta_{\mathcal{I}_g}\|_2$ , where  $\mathcal{I}_g$  is the index set belonging to the  $g^{th}$  group of variables,  $g = 1 \dots G$ . Here each  $g$  is the time index  $t_h \in [t_{-\eta-\delta}, t_{-\delta}]$ , so this group variable selection selects all features of one time in history while reducing some other time points to 0. It has the attractive property that it does variable selection at the temporal group level and is invariant under (group-wise) orthogonal transformations like ridge regression. We note that while there are several other models that could be used for prediction that incorporates the temporal and sequential nature of the data like hidden markov models (HMM) and recurrent neural networks (RNN), the logit model allows us to transparently adjust to the sparsity of data, specially in the absence of a large dataset. For the model with the Group lasso regularization in Equation 3, we set the parameters  $m, l, g$  and 0.3, 0.3 and 0.1 based on a grid search on  $m$

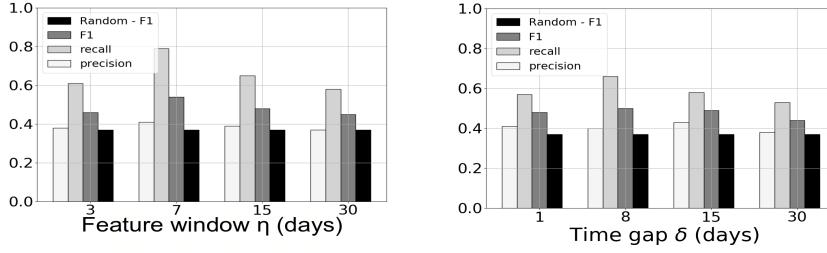


Fig. 13: Classification results for malicious email with feature combination and considering group lasso:  $m=0.3$ ,  $l=0.3$ ,  $g=0.1$ . Refer to Equation 3 for the model used for this prediction.

an  $l$  and keeping  $g$  low so that most time points within a single feature is set to 0 for avoiding overfitting.

We cross validated this model on the 2 hyper-parameters:  $\eta$  and  $\delta$  and we found that while the recall increases for all combinations of hyper-parameters for all features compared to results shown in Figure 11, the precision remains the same across different values of the hyper-parameters. We test on different  $\eta$  keeping  $\delta$  fixed at 8 days and we test on different  $\delta$  keeping  $\eta$  fixed at 7 days. We obtain the best results predicting attacks for the malicious-email type using  $n = 7$  and  $\delta = 8$  days - we get a best F1 value of 0.56 (using  $eta = 7$  days and keeping  $delta$  fixed at 8) using this feature combination model against the best F1 score of 0.53 obtained from using single features without regularization.

## 7 Discussions

As with most machine learning models and setups that attempt binary and multiclass classification including neural networks, the features attributed to the predictions can in most situations explain correlation - the causation needs more controlled studies like visualization by projecting features onto a lower dimensional space, ablation studies or understanding feature importance and using regularization techniques for ensuring sparsity for some features or eliminating redundancy [54]. To this end, we try to investigate whether our framework with the signals from the darkweb discussions correlate to real world events or to other types of attacks. We present 3 controlled studies that show the extent to which the results of our framework are interpretable.

### 7.1 Prediction in High Activity Weeks

One of the main challenges in predicting external threats without any method to correlate them with external data sources like darkweb or any other database is that it is difficult to validate which kinds of attacks are most correlated with

these data sources. To this end, we examine a controlled experiment setup for the *malicious – email* attacks in which we only consider the weeks which exhibited high frequency of attacks compared to the overall timeframe: in our case we consider weeks having more than 5 attacks in test time frame. These high numbers may be due to multiple attacks in one or few specific days or few attacks on all days. The main idea is to see how well does the supervised model perform in these weeks of interest compared to the random predictions with and without prior distribution of attack information. We run the same supervised prediction method but evaluate them only on these specific weeks.

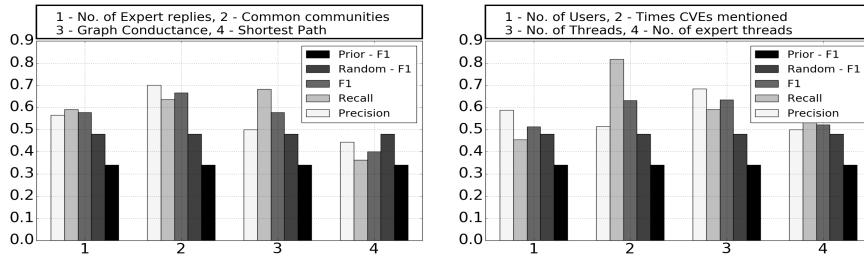


Fig. 14: Classification results for *malicious – email* attacks in high frequency weeks,  $\delta = 7$  days and  $\eta = 8$  days.

From the results shown in Figure 14, we find that the best results were shown by the common communities feature having a precision of 0.7 and a recall of 0.63 and an F1 score of 0.67 compared to the random (no priors) F1 score of 0.48 and a random (with priors) F1 score of 0.34 for the same time parameters. Among the statistics measures, we obtained a highest F1 score of 0.63 for the vulnerability mentions feature. Additionally, we find unlike the results over all the days, for these specific weeks, the model achieves high precision while maintaining comparable recall emphasizing the fact that the number of false positives are also reduced during these periods. This empirically suggests that for weeks that exhibit huge attacks, looking at Darwkeb sources for vulnerability mentions and the network structure analytics can definitely help predict cyber attacks.

## 7.2 Real World Attacks

In order to assess whether the features and the learning model are predictive of *vulnerability exploitation based cyber attack* incidents in the real world, we manually collected one case of vulnerability exploitation that led to real world attacks and which had discussions on the darkweb associated with those vulnerabilities. Since our main evaluations were reported on the malicious email incidents and as mentioned before, the *malicious-email* events are caused by

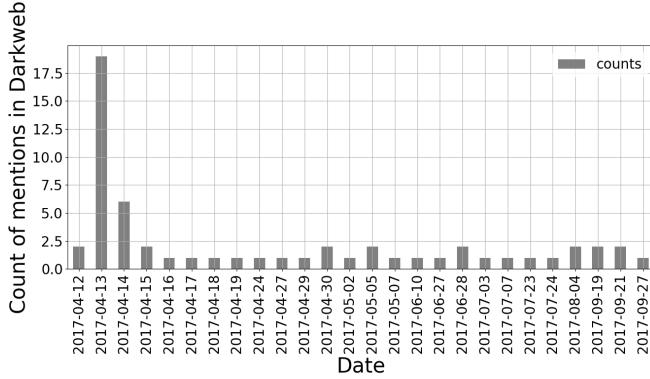


Fig. 15: Lifecycle of darkweb forum mentions of the vulnerability CVE-2017-0199.

malicious email attachments which when downloaded could cause a malicious script to run and execute its code thus intruding the host systems.

*CVE-2017-0199.* This vulnerability is exploited through malicious Microsoft Office RTF documents that allows a malicious actor to download and execute a Visual Basic Script when the user opens the document containing the exploit. As reported in several documents<sup>9</sup>, the document can be sent through an email or a link attachment and therefore is an example of malicious-email breach. This vulnerability has a CVS severity score of 7.8 which is considered high by NIST<sup>10</sup>. There were reports of systems being exploited several months even following the patched date of this vulnerability. In this respect, this vulnerability captured a lot of attention due to the widespread damage that it created. The lifecycle of that vulnerability in the darkweb is shown in Figure 15.

Although Microsoft released the patch on April 11, 2017<sup>11</sup>, discussions started as early as April 12 on the darkweb and there were 18 discussions mentioning the vulnerability on April 13, 2017. When we looked at the content of the discussions on April 13, 2017, we found that most of the discussions surrounding users trying to execute the exploit - whether with malicious intentions or not is a research of sentiment analysis which is also conducted in this domain [55, 56]. When we looked at the attacks in the same and following weeks from Armstrong's malicious email incidents dataset, we found that the first attack occurred on April 13, 2018 and in the following week there were attacks on 3 consecutive days April, 26, 27 and 28 as shown in Figure 16(b).

<sup>9</sup> <https://www.fireeye.com/blog/threat-research/2017/04/cve-2017-0199-hta-handler.html>, <https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-0199>

<sup>10</sup> <https://nvd.nist.gov/vuln/detail/CVE-2017-0199>

<sup>11</sup> <https://blog.talosintelligence.com/2017/04/cve-2017-0199.html>

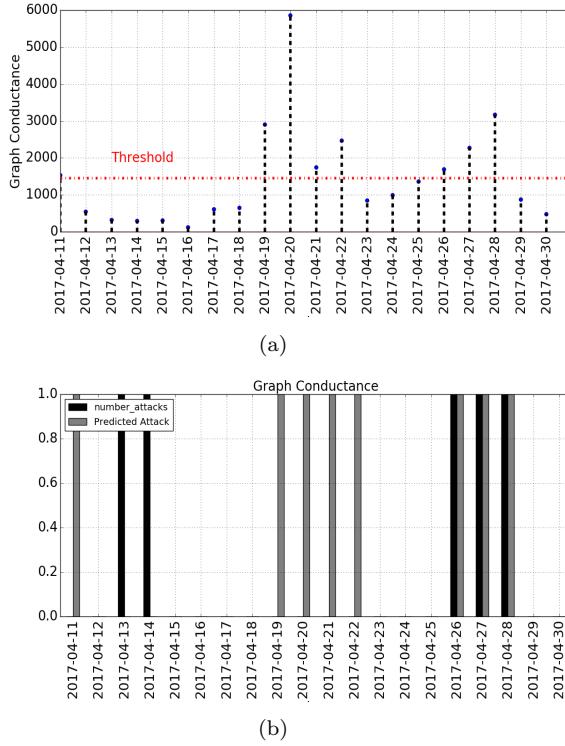


Fig. 16: (a) The graph conductance measure plotted for the weeks from April 11 2017 to April 2017. The red line denotes the threshold  $\delta_\alpha$  above which an anomaly is flagged for that day. (b) The actual attack vector and the predicted attacks in the same weeks.

The period contained a total of 5 days of reported malicious-email incidents in the span of 20 days considered.

We use  $\eta=7$  days, and  $\tau=8$  days for the features (the same parameters used in the previous experiments) and we set  $\zeta=7$ , that is we flag a day  $t$  as an anomaly if  $\mathcal{N}(t) \geq 1$ , or in other words if there is at least one anomaly flagged in the time period  $[t_{-\eta-\delta}, t_{-\delta}]$ . For setting the thresholds that captures whether a particular day has an anomaly in terms of the feature values, we kept the threshold to the mean of the feature values obtained from the training dataset for the respective features. Here we show the feature *Graph Conductance* for the weeks in Figure 16(a), the red line denoting the mean of the training data. We flag any day  $t$  as having an anomaly if the graph conductance on that day crosses the red line. This setup was able to predict the attacks on days April 26, 27 and 28 successfully while missing the attacks on April 13 and April 14. This led to a precision of 0.26 and recall of 0.6 and an F1 score of 0.46 in those 20 days. We have two important observations: first

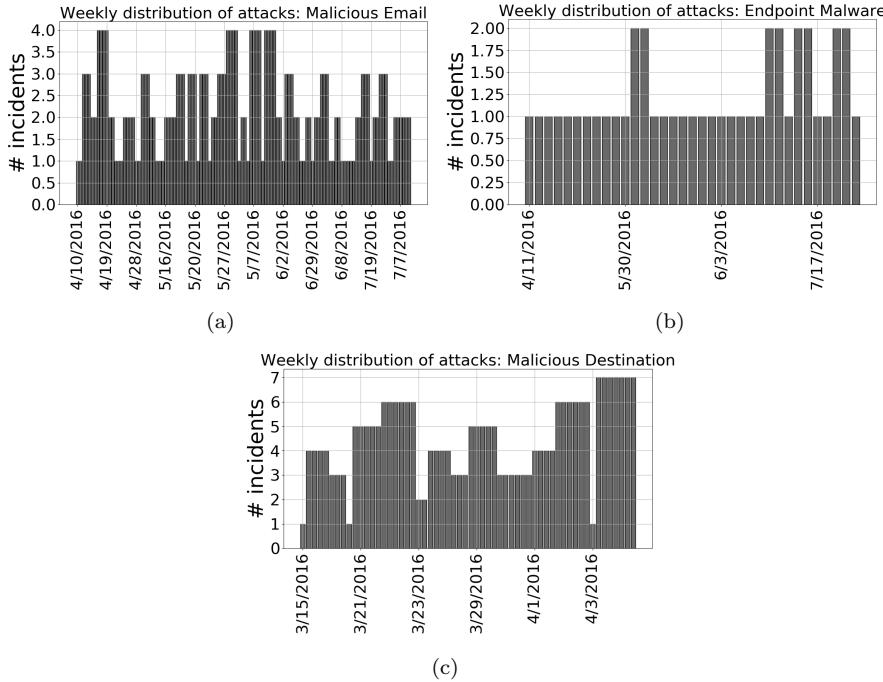


Fig. 17: Weekly occurrence of security breach incidents for Dexter of different types (a) Malicious email (b) Endpoint Malware (c) Malicious destination

it is clear that the predicted attacks on the 3 days were due to the anomalies raised in the previous 2 weeks as shown in Figure 16(b) and secondly, although the CVE mentions shown in Figure 15 does not show any spikes on April 19, 20, 21, 22 and our feature anticipated some anomaly on those days which caused the alerts in the following weeks.

### 7.3 Experiments with another security breach dataset

One of the reasons behind using Armstrong dataset as our ground truth data is the length of the time frame over which the attack data was available - not just the number of attack cases reported (one could have a lot of attack cases reported for only a few days). Since we are attempting a binary classification problem, the more spread the attacks are, the more training point we have for our models and test points for evaluation. However, as a complementary experiment on the learnability of the model parameters specific to companies, we test the prediction problem on a dataset of security incidents from another company named Dexter. As shown in Figure 1, the distribution of attacks over time is different for the events. We observe that compared to the Armstrong dataset, the time span for which the attack ground truth data is available is

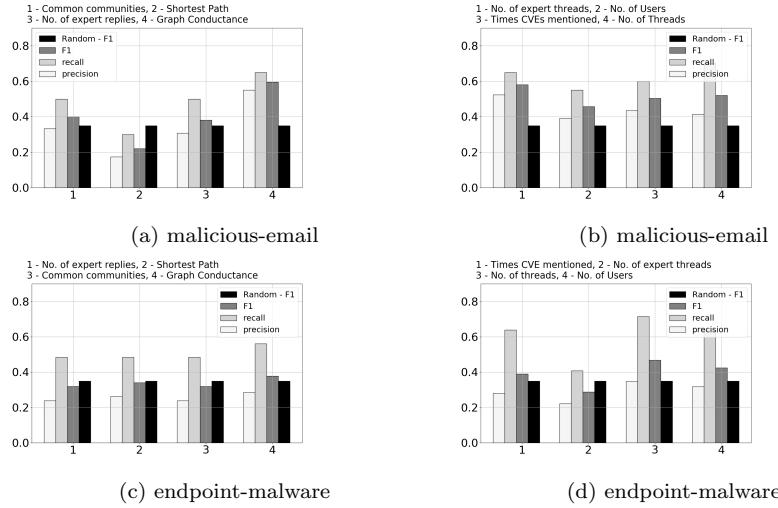


Fig. 18: Classification results on Dexter events for the features considering the supervised model:  $\delta = 7$  days,  $\eta = 8$  days.

much shorter - we obtained around 5 months of attack data for the 3 events shown in Figure 17, starting from April 2016 to August 2016. We have 58 distinct days with at least one incident tagged as *malicious-destination*, 35 distinct days tagged as *endpoint-malware* and 114 distinct days for *malicious-email* events. We had a total of 565 incidents (not distinct days) over a span of 5 months that were considered in our study which is twice the number reported for Armstrong. However, compared to the data spread over 17 months obtained from Armstrong, we have only 4 months to train and test using Dexter data.

We use the same attack prediction framework for predicting the attacks on Dexter, the results of which are shown in Figure 18 - we obtain the best F1 score of 0.6 on the malicious email attacks using the graph conductance measure and an F1 score of 0.59 using the expert threads statistics forum metadata feature (refer to Table 2) against a random F1 score of 0.37. This suggests that the network features which on how experts reply to posts from regular users can be useful in obtaining improved results over other features which do not consider this reply path structure.

## 8 Conclusions and Future Work

In this study, we attempt to empirically argue whether the reply network structure from the darkweb discussions could be leveraged to predict external enterprise threats. We try to leverage the network and interaction patterns in the forums to understand the extent to which they can be used as useful indicators. Our method achieved a best F1 score of 0.53 for one type of attacks against class imbalanced attack data using Logistic Regression models while

being able to maintain high recall. Using an unsupervised anomaly detector, we are able to achieve a maximum AUC of 0.69 by leveraging the network structure. The main premise of this work is based on using two different datasets to correlate attacks and user interactions - the limitations clearly lie in being precisely able to infer the path to the attack through discussions. This would require some additional mechanisms on leveraging the content to check whether the discussions catered to a particular exploit that caused the attack. But we believe that our framework caters to the general understanding of how user interaction patterns can be mined using attributes related to vulnerabilities and how they can be leveraged to create a framework for attack prediction.

### Acknowledgment

Some of the authors are supported through the AFOSR Young Investigator Program (YIP) grant FA9550-15-1-0159, ARO grant W911NF-15-1-0282, and the DoD Minerva program grant N00014-16-1-2015.

### References

1. Liu, Y., Sarabi, A., Zhang, J., Naghizadeh, P., Karir, M., Bailey, M., and Liu, M. (2015, August). Cloudy with a Chance of Breach: Forecasting Cyber Security Incidents. In USENIX Security Symposium (pp. 1009-1024).
2. Thonnard, Olivier, et al. "Are you at risk? Profiling organizations and individuals subject to targeted attacks." International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2015.
3. Xu, Jennifer, and Hsinchun Chen. "The topology of dark networks." Communications of the ACM 51.10 (2008): 58-65.
4. Samtani, Sagar, Ryan Chinn, and Hsinchun Chen. "Exploring hacker assets in underground forums." Intelligence and Security Informatics (ISI), 2015 IEEE International Conference on. IEEE, 2015.
5. Almukaynizi, Mohammed, et al. "Proactive identification of exploits in the wild through vulnerability mentions online." Cyber Conflict (CyCon US), 2017 International Conference on. IEEE, 2017.
6. Almukaynizi, Mohammed, et al. "Predicting cyber threats through the dynamics of user connectivity in darkweb and deepweb forums." ACM Computational Social Science.. ACM, 2017.
7. Liu, Yang, et al. "Predicting cyber security incidents using feature-based characterization of network-level malicious activities." Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics. ACM, 2015.
8. Bilge, Leyla, Yufei Han, and Matteo Dell'Amico. "RiskTeller: Predicting the Risk of Cyber Incidents." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017.
9. Bilge, Leyla, and Tudor Dumitras. "Before we knew it: an empirical study of zero-day attacks in the real world." Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012.
10. Allodi, Luca. "Economic factors of vulnerability trade and exploitation." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017.
11. Edkrantz, Michel, Staffan Truvé, and Alan Said. "Predicting vulnerability exploits in the wild." Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on. IEEE, 2015.

12. Sabottke, Carl, Octavian Suciu, and Tudor Dumitras. "Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits." USENIX Security Symposium. 2015.
13. Khandpur, Rupinder Paul, et al. "Crowdsourcing cybersecurity: Cyber attack detection using social media." Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, 2017.
14. Sapienza, Anna, Sindhu Kiranmai Ernala, Alessandro Bessi, Kristina Lerman, and Emilio Ferrara. "DISCOVER: Mining Online Chatter for Emerging Cyber Threats." In Companion of the The Web Conference 2018 on The Web Conference 2018, pp. 983-990. International World Wide Web Conferences Steering Committee, 2018.
15. Grier, Chris, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J. Dietrich, Kirill Levchenko, Panayiotis Mavrommatis et al. "Manufacturing compromise: the emergence of exploit-as-a-service." In Proceedings of the 2012 ACM conference on Computer and communications security, pp. 821-832. ACM, 2012.
16. Herley, Cormac, and Dinei Florêncio. "Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy." Economics of information security and privacy. Springer, Boston, MA, 2010. 33-53.
17. Allodi, Luca, Marco Corradin, and Fabio Massacci. "Then and now: On the maturity of the cybercrime markets the lesson that black-hat marketeers learned." IEEE Transactions on Emerging Topics in Computing 4.1 (2016): 35-46.
18. Yip, Michael, Nigel Shadbolt, and Craig Webber. "Why forums?: an empirical analysis into the facilitating factors of carding forums." Proceedings of the 5th Annual ACM Web Science Conference. ACM, 2013.
19. Sood, Aditya K., Rohit Bansal, and Richard J. Enbody. "Cybercrime: Dissecting the state of underground enterprise." Ieee internet computing 17.1 (2013): 60-68.
20. Shakarian, Jana, Andrew T. Gunn, and Paulo Shakarian. "Exploring malicious hacker forums." Cyber Deception. Springer, Cham, 2016. 259-282.
21. Kotenko, Igor, and Mihail Stepashkin. "Analyzing vulnerabilities and measuring security level at design and exploitation stages of computer network life cycle." International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security. Springer, Berlin, Heidelberg, 2005.
22. Colbaugh, Richard, and Kristin Glass. "Proactive defense for evolving cyber threats." Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on. IEEE, 2011.
23. Liu, Yang, et al. "Predicting cyber security incidents using feature-based characterization of network-level malicious activities." Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics. ACM, 2015.
24. Phillips, Elizabeth, et al. "Extracting social structure from darkweb forums." (2015): 97-102.
25. L'huillier, Gastón, Hector Alvarez, Sebastián A. Ríos, and Felipe Aguilera. "Topic-based social network analysis for virtual communities of interests in the dark web." ACM SIGKDD Explorations Newsletter 12, no. 2 (2011): 66-73.
26. Sarkar, Soumajyoti, et al. "Predicting enterprise cyber incidents using social network analysis on the darkweb hacker forums." arXiv preprint arXiv:1811.06537 (2018).
27. Haslebacher, Andreas, Jeremiah Onaolapo, and Gianluca Stringhini. "All your cards are belong to us: Understanding online carding forums." Electronic Crime Research (eCrime), 2017 APWG Symposium on. IEEE, 2017.
28. Miller, Charlie. "The legitimate vulnerability market: Inside the secretive world of 0-day exploit sales." In Sixth Workshop on the Economics of Information Security. 2007.
29. Pfleeger, Charles P., and Shari Lawrence Pfleeger. Security in computing. Prentice Hall Professional Technical Reference, 2002.
30. Goyal, Palash, et al. "Discovering Signals from Web Sources to Predict Cyber Attacks." arXiv preprint arXiv:1806.03342 (2018).
31. Okutan, Ahmet, Shanchieh Jay Yang, and Katie McConky. "Forecasting cyber attacks with imbalanced data sets and different time granularities." arXiv preprint arXiv:1803.09560 (2018).
32. Tang, John, Mirco Musolesi, Cecilia Mascolo, and Vito Latora. "Temporal distance metrics for social network analysis." In Proceedings of the 2nd ACM workshop on Online social networks, pp. 31-36. ACM, 2009.

33. Ferrara, E., Varol, O., Davis, C., Menczer, F. and Flammini, A., 2016. The rise of social bots. *Communications of the ACM*, 59(7), pp.96-104.
34. Nagaraja, Shishir. "Anonymity in the wild: Mixes on unstructured networks." In International Workshop on Privacy Enhancing Technologies, pp. 254-271. Springer, Berlin, Heidelberg, 2007.
35. Danezis, George, and Prateek Mittal. "SybilInfer: Detecting Sybil Nodes using Social Networks." In NDSS, pp. 1-15. 2009.
36. Nagaraja, Shishir, Prateek Mittal, Chi-Yao Hong, Matthew Caesar, and Nikita Borisov. "BotGrep: Finding P2P Bots with Structured Graph Analysis." In USENIX Security Symposium, vol. 10, pp. 95-110. 2010.
37. Randall, Dana. "Rapidly mixing Markov chains with applications in computer science and physics." *Computing in Science and Engineering* 8.2 (2006): 30-41.
38. Chierichetti, Flavio, Silvio Lattanzi, and Alessandro Panconesi. "Rumour spreading and graph conductance." Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2010.
39. Tang, John, Mirco Musolesi, Cecilia Mascolo, and Vito Latora. "Temporal distance metrics for social network analysis." In Proceedings of the 2nd ACM workshop on Online social networks, pp. 31-36. ACM, 2009.
40. Yang, Zhao, René Algesheimer, and Claudio J. Tessone. "A comparative analysis of community detection algorithms on artificial networks." *Scientific reports* 6 (2016): 30750.
41. Veeramachaneni, Kalyan, Ignacio Arnaldo, Vamsi Korrapati, Constantinos Bassias, and Ke Li. "AI2: training a big data machine to defend." In 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), pp. 49-54. IEEE, 2016.
42. Xu, Tingyang, Jiangwen Sun, and Jinbo Bi. "Longitudinal lasso: Jointly learning features and temporal contingency for outcome prediction." Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015.
43. Tibshirani, Robert, and Xiaotong Suo. "An ordered lasso and sparse time-lagged regression." *Technometrics* 58, no. 4 (2016): 415-423.
44. Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41.3 (2009): 15.
45. Hodge, Victoria, and Jim Austin. "A survey of outlier detection methodologies." *Artificial intelligence review* 22.2 (2004): 85-126.
46. Lakhina, Anukool, Mark Crovella, and Christophe Diot. "Diagnosing network-wide traffic anomalies." In ACM SIGCOMM Computer Communication Review, vol. 34, no. 4, pp. 219-230. ACM, 2004.
47. Huang, Ling, XuanLong Nguyen, Minos Garofalakis, Michael I. Jordan, Anthony Joseph, and Nina Taft. "In-network PCA and anomaly detection." In Advances in Neural Information Processing Systems, pp. 617-624. 2007.
48. Akoglu, L., Tong, H. and Koutra, D., 2015. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29(3), pp.626-688.
49. Shlens, Jonathon. "A tutorial on principal component analysis." arXiv preprint arXiv:1404.1100 (2014).
50. Soule, Augustin, Kavé Salamatian, and Nina Taft. "Combining filtering and statistical methods for anomaly detection." Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement. USENIX Association, 2005.
51. Rekšņa, Toms. Complex Network Analysis of Darknet Black Market Forum Structure. MS thesis. 2017.
52. Meier, Lukas, Sara Van De Geer, and Peter Bühlmann. "The group lasso for logistic regression." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.1 (2008): 53-71.
53. Zhang, Daoqiang, Jun Liu, and Dinggang Shen. "Temporally-constrained group sparse learning for longitudinal data analysis." International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, Heidelberg, 2012.
54. Ribeiro, M.T., Singh, S. and Guestrin, C., 2016, August. Why should i trust you?: Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144). ACM.

**Algorithm 2:** Computing the time series function  $\mathcal{T}$ 


---

**Input:** Forum posts  $P^f$  for forum  $f$ , time spans  $\Gamma = \{\tau_1, \dots, \tau_k\}$ ,  
 $\mathcal{H} = \{H_{\tau_1}, \dots, H_{\tau_k}\}$

**Output:** Time series function  $\mathcal{T}^f$  mapping the points in  $\Gamma$  to a real value.

```

1  for each  $\tau$  in  $\Gamma$  do
2     $G_{H_\tau} \leftarrow Create(P^f, H_\tau)$ ; // create the historical network using posts
      from time span  $H_\tau$ 
3    for each time index  $t$  in  $\tau$  do
4       $G_t \leftarrow Create(P^f, t)$ ; // create the current network using posts from
        time span  $t$ 
5       $G_{H_\tau, t} \leftarrow Merge(G_{H_\tau}, G_t)$ ; // Create the auxiliary network for  $t$ 
6       $\mathcal{T}^f[t] \leftarrow$  Feature value for time  $t$  considering  $G_{H_\tau, t}$ ;
7    end
8  end
9  return  $\mathcal{T}^f$ 
```

---

55. Al-Rowailly, Khalid, Muhammad Abulaish, Nur Al-Hasan Haldar, and Majed Al-Rubaian. "BiSAL-A bilingual sentiment analysis lexicon to analyze Dark Web forums for cyber security." *Digital Investigation* 14 (2015): 53-62.
56. Chen, Hsinchun. "Sentiment and affect analysis of dark web forums: Measuring radicalization on the internet." *Intelligence and Security Informatics*, 2008. ISI 2008. IEEE International Conference on. IEEE, 2008.

**Appendix**

The outline for the algorithm for creating the social graph  $G$  has been described in Algorithm 2.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/335649598>

# FLOSS FAQ chatbot project reuse: how to allow nonexperts to develop a chatbot

Conference Paper · August 2019

DOI: 10.1145/3306446.3340823

---

CITATIONS

12

READS

781

2 authors, including:



Carla Rocha

University of Brasília

21 PUBLICATIONS 259 CITATIONS

SEE PROFILE

# FLOSS FAQ chatbot project reuse - how to allow nonexperts to develop a chatbot

Arthur R. T. de Lacerda\*

Carla S. R. Aguiar\*

arthurrtl@gmail.com

caguiar@unb.br

UnB Faculty in Gama - University of Brasilia  
Brasilia, Brazil

## ABSTRACT

FAQ chatbots possess the capability to provide answers to frequently asked questions of a particular service, platform, or system. Currently, FAQ chatbot is the most popular domain of use of dialog assistants. However, developing a chatbot project requires a full-stack team formed by numerous specialists, such as dialog designer, data scientist, software engineer, DevOps, business strategist and experts from the domain, which can be both time and resources consuming. Language processing can be particularly challenging in languages other than English due to the scarcity of training datasets.

Most of the requirements of FAQ chatbots are similar, domain-specific, and projects could profit from Open Source Software (OSS) reuse. In this paper, we examine how OSS FAQ chatbot projects can benefit from reuse at the project level (black-box reuse). We present an experience report of a FLOSS FAQ chatbot project developed in Portuguese to an e-government service in Brazil. It comprises of the chatbot distribution service, as well as for analytics tool integrated and deployed on-premises. We identified assets that could be reused as a black-box and the assets that should be customized for a particular application. We categorized these assets in architecture, corpus, dialog flows, machine learning models, and documentation. This paper discusses how automation, pre-configuration, and templates can aid newcomers to develop chatbots in Portuguese without the need for specialized skills required from tools in chatbot

\*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*OpenSym '19, August 20–22, 2019, Skövde, Sweden*

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6319-8/19/08...\$15.00

<https://doi.org/10.1145/3306446.3340823>

architecture. Our main contribution is to highlight the issues non-English FAQ chatbots projects will likely face and the assets that can be reused. It allows non-chatbot experts to develop a quality-assured OSS FAQ chatbot in a shorter project cycle.

## CCS CONCEPTS

- Computing methodologies → Discourse, dialogue and pragmatics;
- Software and its engineering → Open source model; Reusability.

## KEYWORDS

FLOSS, Open source, OSS, FLOSS FAQ chatbot, Black-Box reuse, Portuguese chatbot, experience report, e-government, conversational agents.

## ACM Reference Format:

Arthur R. T. de Lacerda and Carla S. R. Aguiar. 2019. FLOSS FAQ chatbot project reuse - how to allow nonexperts to develop a chatbot. In *The 15th International Symposium on Open Collaboration (OpenSym '19), August 20–22, 2019, Skövde, Sweden*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3306446.3340823>

## 1 INTRODUCTION

Chatbots are conversational software agents that must have natural language processing (NLP) to understand users intentions [18] and has become increasingly popular. They conduct conversations with humans, integrating services, users, and communication channels. Examples of chatbot applications are notification, FAQ [11, 23], virtual host [15], and personalized assistants [17].

The most common type of conversational agent is FAQ chatbots, which possesses the capability to provide answers to frequently asked questions of a particular service, platform or system [23]. A consequence of using FAQ chatbots in organizations services is the decrease of the demand on other communication channels, such as calls and e-mails [10].

OSS chatbots allow to keep all conversation data on-premises and facilitate their compliance with data regulations, such as the General Data Protection Regulation (GDPR) [9, 28]. They can benefit from collaborative modeling [21], and other bot

data, such as personality, vocabulary, and chit chat datasets. Chatbot architecture, composed by the integration of both building/creating a chatbot and distributing/messaging service, algorithms, and analytics is transparent in FLOSS. However, most of the FLOSS chatbot frameworks are domain-agnostic, where they handle one specific layer of the chatbot architecture and they leave the developer to configure algorithms and parameters for their given application. The documentation found in these communities is mostly technical, done by and to experts of the field. It thus discourages nonexperts practitioners to choose which FLOSS chatbot framework to reuse for a specific application reliably.

In this work, we investigate the assets necessary to FLOSS FAQ chatbot project to be reused as a black-box. We chose the application domain of FAQ to restrain the scope since the requirements specificity affects software positively reuse success [20]. Not only code reuse, but also architecture, content (dialogues, intentions), and machine learning conversation models. We present an experience report, where we conducted a project of a FLOSS FAQ chatbot developed during 15 months of a government-academia collaboration in Brazil, with best practices based on FLOSS ecosystems, agile and DevOps. Our main contribution is to identify and trace assets to maximize reuse in this application domain, regarding toolset, architecture, algorithms, training dataset, and documentation. We collect and analyze data from the project repository and data from analytics. We contribute to guide nonexperts practitioners to develop their FAQ chatbot in reduced project time and effort.

## 2 RELATED WORK

According to Lebeuf in [11], a chatbot project is composed of distribution and creation services. Distribution services are employed to manage the conversation data, it is the interface between the chatbot and the user, while creation services are used to develop the chatbot knowledge and behavior. Optionally, the business analytics service helps the improvement and evolution of the chatbot. The present work is focused on OSS chatbot distribution services.

The distribution service itself is composed of the following components [8]: Natural Language Understanding (NLU), communication channels, voice user interfaces, dialog management.

A vast number of chatbot frameworks is available in OSS, such as Botkit [11], Rasa [2], Botpress [3], among others. Typically, OSS chatbot frameworks are designed for developers and specialists, and an issue is to enable nonexperts to use them as a black box [28]. An exception is botpress, a complete chatbot architectural integrated solution, with a giving user interface that guides nonexperts from bot content creation to deployment. Limitations of this solution it is being both agnostic-domain and restrains the algorithms

used. Dialog management that handles dialog context and decides the next action for the agent to take have drifted from pure rule-based to data-driven in recent years [8, 19].

Code reuse allows for previously tested and quality-assured code to be implemented in another system and it provides benefits regarding simply adding and enhancing system features [22]. In chatbot projects, both white-box and black-box reuse are common. However, most of the research done focus on the reuse of training datasets [7, 14, 19].

One crucial part of the chatbot is to understand user messages. According to Liu et al. [13], Natural Language Understanding Services perform similarly in both OSS and proprietary solutions. Another grand challenge is the effectiveness of chatbots in non-English speaking countries [14]. When the objective is to build a chatbot in other languages than English, dialogue datasets are the most significant difficulty [7], but it still possible as shown in [24], a German chatbot was built to guide an idea submission process. ParlAI is an open-source platform that aims to minimize this dataset issue by providing a unified framework for sharing, training, and testing dialog models [16]. They aim to reinforce reuse of training datasets by sharing a repository of the corpus, utterances, and machine learning models. Another alternative is the adoption of crowdsourcing to write utterances. Paetzel et al. in [19] evaluate how untrained crowd workers (crowdsourcing) can scale chatbots contents and still maintain coherence in the chatbot personality, affective behavior, and vocabulary.

## 3 THE EXPERIENCE REPORT

The project to develop an FAQ chatbot is a partnership between government and academia started in 2017 [1], and it is still ongoing. The Ministry of Citizenship decided to join the University of Brasília (UnB) to develop a FLOSS FAQ chatbot to help citizens to understand better their law of incentive to the culture and to answer the frequently asked questions of its service. However FAQ chatbot presence is growing in private sectors, it is still rare in government agencies [10].

One of the significant project requirement that it should be based on existing FLOSS, and the Ministry technical staff should handle its evolution and maintenance. The technical staff is composed mainly by software engineers, journalists, and experts in cultural law, and none of them had any prior experience in developing chatbots. Therefore, project documentation, configurations, and automation should be designed to shorten the team learning curve and to render unnecessary the need for chatbot specialists in the team. The main characteristics of this project are depicted in Table 1.

**Table 1. Characteristics of our FLOSS chatbot project.**

Team Members from all needed expertise	14 members
Releases in the last 12 months	14 releases
Number of Intentions	72 annotated intents
Number of Utters	147 utters
Original Size of the FAQ	35 questions

Throughout the project, we develop incrementally and employ a set of best practices based on FLOSS, agile and DevOps values, with lessons learned from previous government-academia collaboration projects [27]. We focused on facilitating reuse by the design of a modular architecture, modular corpus, and integration of highly dynamic OSS projects, up-to-date documentation, containerization, and heavy use of continuous integration tools to orchestrate several automated actions that, together, enabled the deployment pipeline.

Based on our practical experience, we present the FLOSS chatbot for FAQ e-government services, the lessons learned, and how the black-box reuse can reduce the time of new projects of FAQ chatbots to mature.

#### 4 THE CHATBOT PROJECT

In the following sections, we present our solution and how significant decisions were made toward reuse. While throughout the project, we had experts in the teams or developed the necessary abilities in-house, this work serves as an experience report that enables black-box reuse to empower nonexperts in developing FAQ chatbots.

##### Full-stack Team

Choosing the most appropriated frameworks, tools, algorithms and dialogue models require expertise in fields such as machine learning, DevOps, language, User Experience (UX), project management, business strategy, among others [5].

A typical full-stack bot team can be divided into organization, development and beta test members, and each group has one particular involvement with the project (Figure 1). It is essential to have context specialist to give the necessary content and business specialists to guide towards the business goals. Also, it is crucial to have IT specialists involved.

In the present project, the team was formed by undergraduate interns, IT professionals and professors. The DevOps role is necessary to manage automation, services integration, and continuous deploy configuration. The UX Specialists will plan all the dialog flows, chatbot personality, and vocabulary.

Finally, Data Scientists choose the techniques to process natural language, dialogues and calibrate the hyperparameters of such models.

Beta testers are essential to validate new features of the chatbot, provide feedback before it is deployed into production.

Although we have established this team structure, an essential requirement was to guarantee that a "traditional" team structure could do the maintenance, as depicted in Figure 1 (D),(E),(F). The following sections will detail how it was achieved with a focus on black-box reuse.

##### Overview

The complete overview of our project is depicted in Figure 2, and every component is OSS and integrated. It guarantees that the entire solution is on-premises, and how data is processed, stored, and distributed are transparent and customized.

Figure 2(A) represents the Communication Channel, and it is the interface where the user will send messages to the chatbot, and it will receive answers. Rocket.Chat<sup>1</sup> is a communication tool used. It enables a higher degree of automation, and the capability to store all conversations. Figure 2(B) is the core of the chatbot, where both natural language understanding and dialog management is performed. Rasa NLU and Rasa Core<sup>2</sup> are our distribution services. They implement state-of-art NLP algorithms and aim to bridge the gap between research and application. The quality of annotated intents is asserted by analyzing the dataset in Jupyter Notebooks. In Figure 2(C) is displayed the business analytics, the stack elasticseach and kibana provide dashboards to gather all information about the users, like the most asked questions, chatbot access, timestamp, and unmapped questions, which can be interpreted by the Organization and the Chatbot Team. We did not use any creation service, and the entire bot knowledge and behavior were done directly on markdown files of the distribution services.

<sup>1</sup><https://rocket.chat>

<sup>2</sup><https://rasa.com>

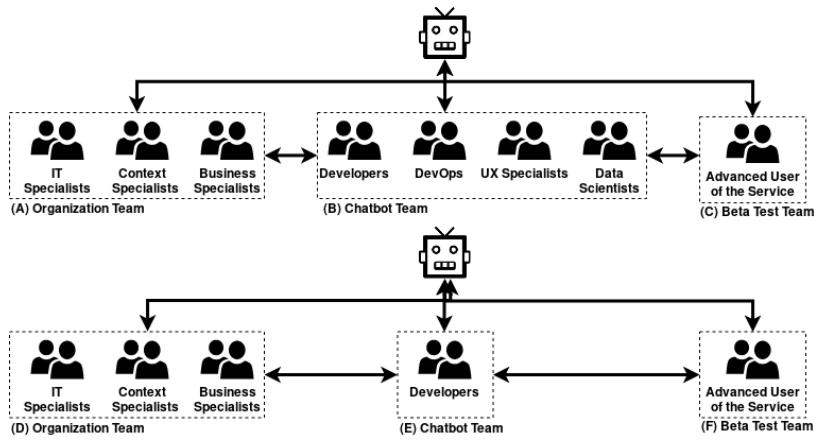


Figure 1: Chatbot team with experts (a),(b),(c) and non experts with project reuse (d),(e),(f).

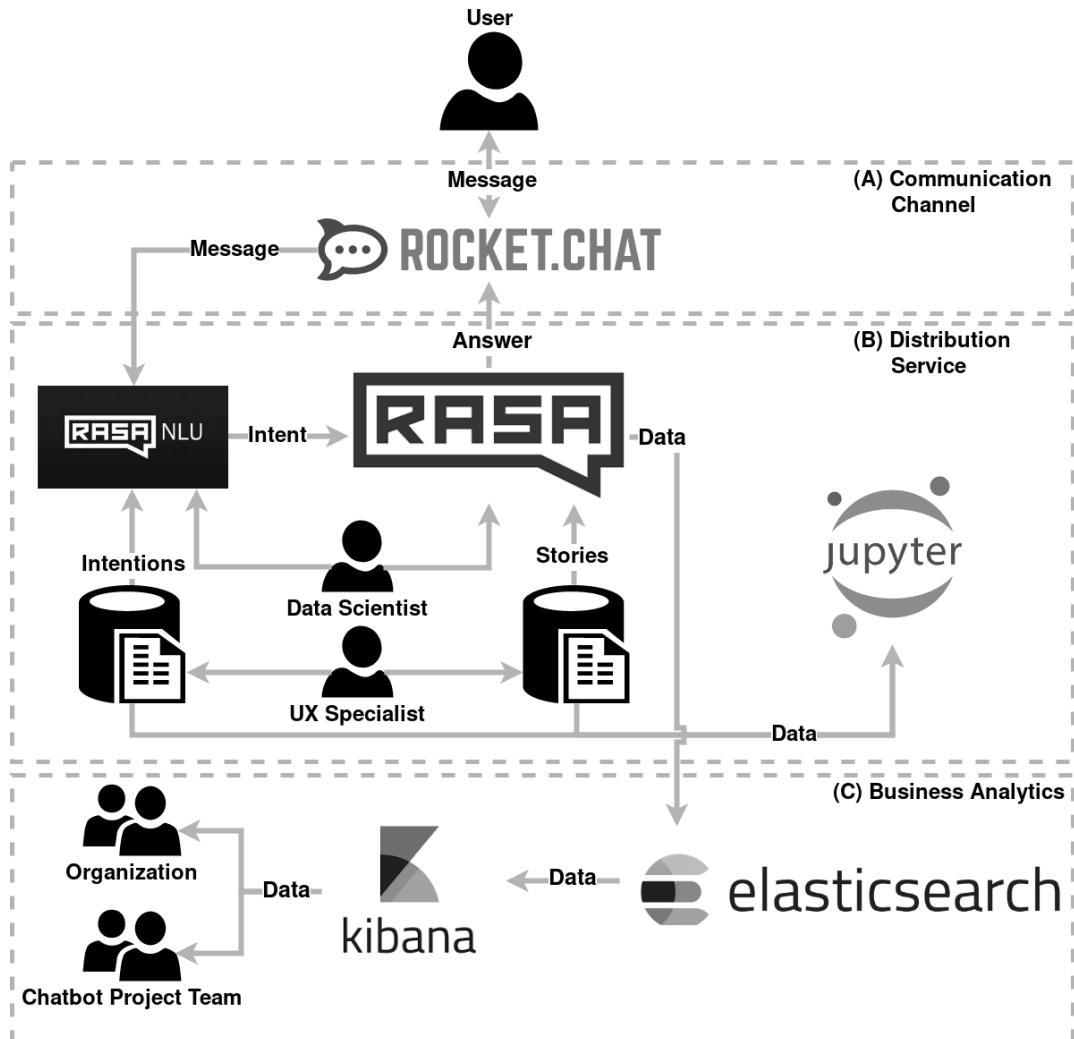


Figure 2: Project overview - technologies, artifacts, and stakeholders.

The automation of the deployment pipeline facilitates reuse of the architecture solution. We employ containerization to each service separately, and configuration files to customize production, homolog, and training environments. Finally, each component of the architecture is loosely coupled and well-encapsulated so that it can be both customized or even replaced.

### **Interaction Design**

Designing the corpus of the chatbot knowledge base and strategy to manage the dialog is time-consuming, hand-authored by UX and interaction designer experts, which does not scale to large domains [19]. While crowdsourcing can help to overcome the problem of scale, it is only possible when it is clear the chatbot personality, vocabulary and writing style.

We addressed two main interaction design challenges: (a) a guideline of how to convert FAQ texts into annotated intents, dialog arcs and dialog flows, and (b) separate chatbot specific dataset content from general dialogue elements, such as chitchat, general intents, fallback strategy. This later could be black-box reused in any FAQ chatbot, and interaction design would be reduced to create training data to the specific FAQ domain, following provided tutorials.

We have documented the entire interaction design process, and we create tutorials, best practices, and guidelines to write annotated intents from FAQs, suggested a personality, vocabulary, and repeated the same procedure to create dialogue contents. Finally, chatbot utterances were separated in files to explicit domain-specific content from general bot behavior.

### **Natural Language Understanding and Dialog Management**

Chatbots are based on machine learning techniques that (a) process natural language from input message from users, (b) identify the message meaning (Natural Language Understanding), and (c) select the most appropriate next action to perform (Dialog management) [8]. The choice of machine learning techniques, their pre and post-processing impact the user experience directly. For example, it influences how well the chatbot identifies user intent on a given language, its ability to infer context, to treat noncooperative behavior, and to manage complex dialog flows [8][10]. Therefore, the proper choice of Natural Language Understanding (NLU), dialog management pipeline and hyper-parameters impose restrictions on what can be done by the chatbot.

FAQ chatbots basically must identify users intents and select the correct action to execute from a closed list. The choice of the most suitable NLU and dialog management techniques depends on the amount of dataset available for training and validation. This dataset is composed by the

list of utterances annotated with intents, entities, language annotated dataset, and dialogues examples [16]. It implies that techniques execution pipeline can be borrowed across projects with similar requirements, like FAQ chatbots, as a form of generalization and reuse.

We have chosen data-driven algorithms that worked well for Brazilian Portuguese language and medium-size utterances (over 1000). For intent classification, we chose the supervised embed model Starspace [2], and we utilize annotated dialogues to train a Long short-term memory (LSTM) network [25]. This pipeline can be reused as a black-box to any FAQ chatbot in Portuguese, where the only configuration necessary is the calibration of the hyperparameters.

To validate the hypothesis of algorithms pipeline reuse, we wrote a tutorial on hyperparameters calibration, and ask a group of software engineering undergraduates, with no prior knowledge in machine learning, to develop FAQ chatbots to the University using the same pipeline. In three months, they were able to develop a chatbot and correctly calibrate the hyperparameters, guided by our tutorials.

### **Analytic and Improvements**

According to Michaud [15], monitoring is an important step while developing a chatbot. Analytic tools track messages, distinguishes guests and users text behavior, and show in dashboards an infinity of graphics that display metrics and data from the actual chatbot use. This information enables the management of the knowledge base and guides the development efforts aligned with the business strategy.

The lesson learned from adding analytics to our chatbot project is the importance of determining the quality of the utterances and dialogue contents [15], in providing accurate data on users real questions and real behavior. This work is unfeasible manually. Metrics guide the development team to improve the quality of the interaction, and they also guide managers to draw data-driven business strategies. Table 2 summarizes some of the data provided by analytics, showing some results from our chatbot data after sixty days in the production environment.

## **5 DISCUSSION**

Throughout this paper, we presented a report of a FLOSS FAQ chatbot project. The objective is to present some of the issues engineers, experts, managers, or researchers will likely be confronted with when developing an OSS FAQ chatbot, some core concerns they should have, and to evidence how they can benefit of black-box reuse of similar project.

Although white-box reuse is an essential practice in Open Source Projects, chatbots can benefit from black-box reuse regarding (a) machine learning models, and techniques pipeline,

**Table 2. Business analytics data collected after 60 days of the chatbot in production.**

Average Number of Messages sent by single user	10.2
Number of users interacting with the chatbot per day	44
Number of Default fallback occurrences in 60 days	72
Average number of times each user asks for help per conversation	2
Messages per day on week days	738
New questions not present in utterances	121
Number of intents added after analysing the data	14

(b) datasets to train both intent and dialog flows, (c) automation, configuration files, and (d) analytics tools and dashboards. Consequently, with proper documentation, tutorials, and guidelines, untrained workers/ nonexperts could build a mature FLOSS FAQ chatbot from scratch in a short period. Table 3 a list of assets that could be reused as black-box, and the ones that should be customized if our project is used to develop a FAQ chatbot in Portuguese in a different context than ours.

A common problem in chatbot projects is to find, to chose, and to configure the most appropriate set of technologies. In OSS, this problem is accentuated, once most projects only provide advanced technical documentation [4], which creates an adoption barrier to nonexperts. Additionally, some of the most advanced OSS frameworks cover only partially a chatbot architecture, and it imposes an integration effort which is also an adoption barrier to nonexperts. Finally, the common dialogue flows like greeting and chit chat are typically re-implemented every new application. Although it is readily available this corpus in English, the same is not correct to other languages. Therefore, all these barriers make a typical FAQ chatbot project to be time-consuming, filled with mistakes, waste, and to deploy immature chatbots in the production environment. We believe these projects errors can shadow the efficacy of FAQ chatbots, and by poor user experience lead by immature chatbots [14].

In 1996, Weizenbaum [26] called Eliza as a program, in 1997 Lieberman [12] called Letizia as a computer agent, but both Eliza and Letizia are classified today as chatbots. Nowadays, there is still some divergence to similar chatbot concepts, such as in [6] that evidences synonyms like conversational systems and virtual agents, chatbot. The lack of a common vocabulary added disturbance to this study. The discussion and definition of these concepts is an opportunity for academic contribution.

This study has a few apparent limitations. Around 5 groups of much more inexperienced teams, composed by undergraduate students, developed a chatbot to answer FAQ of the administration services at the university in 3 months,

reusing ours as presented in the paper. However, this validation should be done to a broader number of FAQ projects, and data about collected to validate our hypotheses fully. Future work is needed to explore more sophisticated approaches to this problem, and compare reuse in contexts other than FAQ chatbots.

## 6 CONCLUSIONS

In this paper, we presented the experiment report of a FLOSS FAQ chatbot project developed in the context of e-government services. We highlight the main challenges encountered when developing a chatbot distribution service and analytics for non-English dialogue agents. Several OSS projects compose the chatbot architecture. Their selection is the first difficulty, and automation is fundamental to integrate these services and enable continuous delivery. The choice of Natural Language Understanding (NLU) and Dialogue Management techniques demands the expertise of data scientists, and we found that the OSS community gives little support to non-English applications. We configured a set of techniques that performed well for the Portuguese Language with small training datasets. We provided tutorials to facilitate the reuse and hyperparameters calibration to different contexts and scopes. These automation, customizations, and tutorials enable to use this project in other FAQ chatbots in black-box reuse.

Finally, we have outlined a set of good practices, tutorials, and documentation to empower untrained workers/ nonexperts to build a FLOSS FAQ chatbot from scratch in a short period for non-English contexts.

## 7 ACKNOWLEDGEMENTS

The authors are indebted to the users and collaborators of this chatbot project for providing invaluable feedback and creating a supportive ecosystem. Special acknowledgment is owed to the Lab team and external contributors. The up-to-date list of contributors may be visited at <https://github.com/lappis-unb/tais/graphs/contributors>.

The Brazilian Ministry of Citizenship has financially supported this research in the TAIS project. The authors are

**Table 3. Black-box reuse in FLOSS FAQ Chatbot - list of assets.**

Activity	Assets to be reused as black-box	Assets to be customized for each application
<b>Interaction Design</b>	Language corpus General dialogue contents Dialogue tutorials of how to convert FAQ file into intents and utters Chatbot personality tutorials	Write intents, utters and dialogue flows to the desired context Follow dialogue tutorials Follow personality tutorials
<b>Architecture</b>	Distribution service Creation service Analytics service Continuous integration Development containers Experimental environment for dataset validation Configuration tutorials Architecture customization tutorials	Follow configuration tutorials Follow architecture customization tutorials
<b>Natural Language Understanding</b>	Technics pipeline to brazilian portuguese NLU hyperparameters calibration tutorial	Follow tutorial to calibrate NLU hyperparameters
<b>Dialog Management</b>	Dialog Management policy techniques Dialog Management policy hiperparameters calibration tutorial	Follow tutorial to calibrate the hyperparameters of the dialogue management policy
<b>Monitoring</b>	Dashboards templates Dashboards customization tutorials documentation	Follow dashboards customization tutorials

grateful for the stimulating collaboration and support from colleagues and partner organization.

## REFERENCES

- [1] Research Advanced Laboratory of Production and Innovation in Software Engineering (LAPPIS/UnB). 2019. Tais - a FLOSS FAQ Chatbot for the Ministry of Culture. <https://github.com/lappis-unb/tais>
- [2] Tom Bocklisch, Joey Faulker, Nick Pawłowski, and Alan Nichol. 2017. Rasa: Open Source Language Understanding and Dialogue Management. *CoRR* (12 2017).
- [3] Botpress. 2019. Botpress - A Chatbot Maker & Development Framework. <https://botpress.io/>
- [4] Noel Carroll, Lorraine Morgan, and Kieran Conboy. 2018. Examining the Impact of Adopting Inner Source Software Practices. In *Proceedings of the 14th International Symposium on Open Collaboration (OpenSym '18)*. ACM, New York, NY, USA, Article 6, 7 pages.
- [5] Jessica Falk, Steven Poulakos, Mubbasis Kapadia, and Robert Sumner. 2018. PICA: Proactive Intelligent Conversational Agent for Interactive Narratives. 141–146.
- [6] B Filipczyk. [n.d.]. Chapter 12 - Success and failure in improvement of knowledge delivery to customers using chatbotâ€”result of a case study in a Polish SME. ([n. d.]), 15.
- [7] Mingkun Gao, Wei Xu, and Chris Callison-Burch. 2015. Cost Optimization in Crowdsourcing Translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2015)*. Denver, Colorado.
- [8] J. Harms, P. Kucherbaev, A. Bozzon, and G. Houben. 2019. Approaches for Dialog Management in Conversational Agents. *IEEE Internet Computing* 23, 2 (March 2019), 13–22.
- [9] Eric von Hippel and Georg von Krogh. 2003. Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science* 14, 2 (March 2003), 209–223.
- [10] P. Kucherbaev, A. Bozzon, and G. Houben. 2018. Human-Aided Bots. *IEEE Internet Computing* 22, 6 (Nov 2018), 36–43.
- [11] C. Lebeuf, M. Storey, and A. Zagalsky. 2018. Software Bots. *IEEE Software* 35, 1 (January/February 2018), 18–23.
- [12] Henry Lieberman. 1997. Autonomous interface agents. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '97*. ACM Press, Atlanta, Georgia, United States, 67–74.
- [13] Xingkun Liu, Arash Eshghi, Paweł Świętajanski, and Verena Rieser. 2019. Benchmarking Natural Language Understanding Services for building Conversational Agents. *arXiv* 1903.05566 (mar 2019), 13.
- [14] R Lowe, N Pow, Iulian Serban, L Charlin, C.-W Liu, and J Pineau. 2017. Training end-to-end dialogue systems with the Ubuntu Dialogue Corpus. *Dialogue and Discourse* 8 (01 2017), 31–65.
- [15] L. N. Michaud. 2018. Observations of a New Chatbot: Drawing Conclusions from Early Interactions with Users. *IT Professional* 20, 5 (Sep. 2018), 40–47.
- [16] Alexander H. Miller, Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, and Jason Weston. 2017. ParlAI: A Dialog Research Software Platform. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (2017).
- [17] Mycroft. 2019. Mycroft AI Open Source Voice Assistant. <https://mycroft.ai>
- [18] M. Nuruzzaman and O. K. Hussain. 2018. A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*. 54–61.
- [19] Maike Paetz, James Kennedy, Ginevra Castellano, and Jill Lehman. 2018. Incremental Acquisition and Reuse of Multimodal Affective Behaviors in a Conversational Agent. In *International Conference on Human-Agent Interaction 2018*. 92–100.
- [20] Maria-Eleni Paschali, Apostolos Ampatzoglou, Stamatia Bibi, Alexander Chatzigeorgiou, and Ioannis Stamelos. 2017. Reusability of open source software across domains: A case study. *Journal of Systems and Software* 134 (2017), 211 – 227.
- [21] S. Perez-Soler, E. Guerra, and J. de Lara. 2018. Collaborative Modeling and Group Decision Making Using Chatbots in Social Networks. *IEEE Software* 35, 6 (November/December 2018), 48–54.

- [22] Ruben Prieto-Diaz. 1993. Status Report: Software Reusability. *Software, IEEE* 10 (06 1993), 61 – 66.
- [23] B. R. Ranoliya, N. Raghuwanshi, and S. Singh. 2017. Chatbot for university related FAQs. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 1525–1530.
- [24] Navid Tavanapour and Eva A C Bittner. 2018. Automated Facilitation for Idea Platforms: Design and Evaluation of a Chatbot Prototype. *Conference: Thirty Ninth International Conference on Information Systems (ICIS)* (2018), 9.
- [25] Vladimir Vlasov, Akela Drissner-Schmid, and Alan Nichol. 2018. Few-Shot Generalization Across Dialogue Tasks. *CoRR* (2018).
- [26] Joseph Weizenbaum. 1966. ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine. *Commun. ACM* 9, 1 (Jan. 1966), 36–45.
- [27] Melissa Wen, Paulo Meirelles, Rodrigo Siqueira, and Fabio Kon. 2018. FLOSS Project Management in Government-Academia Collaboration. In *International Conference on Open Source Systems*. 15–25.
- [28] Mairieli Wessel, Bruno Mendes de Souza, Igor Steinmacher, Igor S. Wiese, Ivanilton Polato, Ana Paula Chaves, and Marco A. Gerosa. 2018. The Power of Bots: Characterizing and Understanding Bots in OSS Projects. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 182 (Nov. 2018), 19 pages.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/368832149>

# Precision in post-correction of annotated corpus 1

Technical Report · February 2023

DOI: 10.13140/RG.2.2.33247.94888

---

CITATIONS  
0

READS  
5

1 author:



Arvi Hurskainen  
University of Helsinki  
59 PUBLICATIONS 199 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Rule-based machine translation between structurally complex languages [View project](#)

## Precision in post-correction of annotated corpus<sup>1</sup>

Arvi Hurskainen  
Department of Languages  
FIN-00014 University of Helsinki, Finland  
[arvi.hurskainen@helsinki.fi](mailto:arvi.hurskainen@helsinki.fi)

### Abstract

It is common that corpus annotation contains mistakes that need to be corrected. There are basically two methods for doing this. One solution is to correct the tagger itself and run the tagging process again. Another solution is to correct only those readings that need correction. In this report I discuss such a case, where the corpus was tagged using a tagging program, and after that several corrections were made either manually or by using correction scripts. As a result, the corpus was considered properly tagged. However, later it was found that there was a commonly occurring verb form, which was not recognised by the tagger in part of the corpus, and it was interpreted as a noun, using a heuristic post-tagger.

It was not feasible to re-analyse the corpus using the corrected analyser, because in that solution the post-processing corrections would be lost. There was a need to correct only those readings, where the mistakes occur and leave other readings intact.

The corpus that I discuss here is Helsinki Corpus of Swahili 2.0 containing 25 million words. The construction of the correction script in this case is not a simple task, because Swahili verbs inflect to both directions, using sequences of prefixes and suffixes. In addition, the noun class system multiplies the number of possible verb forms of each verb.

I will demonstrate phase by phase how the correction script can be constructed.

**Key Words:** *morphological analysis, corpus annotation.*

### 1 Introduction

The Language processing tools are generally defective. The best of them perform remarkably well, but it is common that the analyser makes some irritating mistakes. Although we try to correct the analyser, the language is so complex that mistakes cannot be avoided. Often the mistakes are related to defective tokenisation and such words that the analyser does not recognize. Such mistakes cannot perhaps be totally avoided. A more serious case is such a mistake type that is related to the inflection of the words.

This report concerns the correction of a particular verb form in Helsinki Corpus of Swahili 2.0.<sup>2</sup> In a large part of the corpus text, the analyser did not recognize such verb forms that

---

<sup>1</sup> The report is issued under licence CC BY-NC

<sup>2</sup> <http://urn.fi/urn:nbn:fi:lb-201608301>

included the narrative case, which is marked with the *-ka-* prefix. It is not known how this bug entered the code in the processing phase. Since the analysed corpus went through heavy post-processing routines, both manually and by using various correction scripts, it was not feasible to re-analyse the corpus using the corrected analyser.

I took another approach, which resembles a medical operation, where a curing ingredient is sent to a certain part of the body without manually touching the target, or even without seeing it. Yet my task was different in that instead of a single operation, the script should handle correctly hundreds of thousands different types of cases.

## 2 Phases in constructing the correction script

I will describe below the phases of the construction of the correction script. I will use in the demonstration a set of various types of cases. The cases are extracted from the old version of the corpus (1).

(1)	a. ikawa	ikawa	N	Heur 9/10-SG	{ ikawa }	@<P
	b. tukalazimika	tukalazimika	N	Heur 9/10-SG	{	
	tukalazimika }		@<P			
	c. tukawaeleze	tukawaeleze	N	Heur 9/10-SG	{	
	tukawaeleze }		@SUBJ			
	d. tukayaboresha	tukayaboresha	N	Heur 9/10-SG	{	
	tukayaboresha }		@OBJ			
	e. akamteua	akamteua	N	Heur 9/10-SG	{ akamteua }	
			@SUBJ			
	f. zikatumika	zikatumika	N	Heur 9/10-SG	{ zikatumika	
	}		@OBJ			
	g. yakawanufaisha	yakawanufaisha	N	Heur 9/10-SG	{	
	yakawanufaisha }		@<P			
	h. likaangaliwe	likaangaliwe	N	Heur 9/10-SG	{	
	likaangaliwe }		@OBJ			
	i. ukatarajia	ukatarajia	N	Heur 11-SG	{ ukatarajia }	
			@OBJ			
	j. wakajinunua	wakajinunua	N	Heur 1/2-PL	{	
	wakajinunua }		@OBJ			

We see that if the verb has a prefix *-ka-* it is not recognized by the analyser. Instead, it is interpreted as a noun, because normally verbs get an analysis, while new nouns are constantly introduced to the language. Because the word is unknown, the wordform is copied as a stem. The noun class is guessed on the basis of the prefix and the corresponding tag is added. The default class pair for new nouns is 9/10, and most unknown words were given that class tag. Only in (i) and (f) the tag is different. The gloss in English is within curly braces. We see that it is the copy of the word. The syntactic tag is @SUBJ or @OBJ, and in some cases @<P. All this was done on a shaky basis.

When we start to convert these readings into correct form, we first mark the readings in the corpus, so that, when we run the correction script, we only touch these readings and leave other readings intact.

In the first phase we mark the faulty lines by adding '&' in front of them. We also remove the wrong readings and give some new analysis (2).

(2)

```
&ikawa ika wa V @FMAINV
&tukalazimika tuka lazimika V @FMAINV
&tukawaeleze tuka waeleze V @FMAINV
&tukayaboresha tuka yaboresha V @FMAINV
&akamteua aka mteua V @FMAINV
&zikatumika zika tumika V @FMAINV
&yakawanufaisha yaka wanufaisha V @FMAINV
&likaangaliwe lika angaliwe V @FMAINV
&ukatarajia uka tarajia V @FMAINV
&wakajinunua waka jinunua V @FMAINV
```

We see in (2) that we have tried, by adding a single space after *-ka-*, to identify the point, where the verb stem starts. In some cases, it is correct, but not in such verb forms that also include the object prefix. We also know that the words are verbs in finite form. Therefore, readings are given the syntactic tag @FMAINV.

In the original corpus, tabs are used for separating sections. Because the use of tabs is not convenient in writing rules, I have replaced tabs with double spaces. They can be returned back to tabs later. Note that between the prefix cluster and the stem there is only one space.

Next we separate the object prefixes from the stem (3).

(3)

```
&ikawa ika wa V @FMAINV
&tukalazimika tuka lazimika V @FMAINV
&tukawaeleze tuka +wa eleze V @FMAINV
&tukayaboresha tuka +ya boresha V @FMAINV
&akamteua aka +m teua V @FMAINV
&zikatumika zika +tu mika V @FMAINV
&yakawanufaisha yaka +wa nufaisha V @FMAINV
&likaangaliwe lika angaliwe V @FMAINV
&ukatarajia uka tarajia V @FMAINV
&wakajinunua waka +ji nunua V @FMAINV
```

Now object prefixes are separated from the stem. Also an anchor '+' is added in front of the prefix, so that further processing becomes easier. However, we see that the word *zikatumika* is wrongly interpreted. It should be *zika tumika*. Mistakes such as this are corrected using suitable scripts.

In the next phase, we mark the verb stems by adding a colon ':' in front of the stem. This is done for making sure that when English glosses are added to the readings, the stem is clearly identified (4).

(4)

```
&ikawa ika :wa V @FMAINV
&tukalazimika tuka :lazimika V @FMAINV
&tukawaeleze tuka +wa :eleze V @FMAINV
&tukayaboresha tuka +ya :boresha V @FMAINV
&akamteua aka +m :teua V @FMAINV
&zikatumika zika :tumika V @FMAINV
&yakawanufaisha yaka +wa :nufaisha V @FMAINV
&likaangaliwe lika :angaliwe V @FMAINV
&ukatarajia uka :tarajia V @FMAINV
&wakajinunua waka +ji :nunua V @FMAINV
```

The stems are now separated from prefixes, but they still have suffixes, which affect the import of English glosses. Therefore, they must be converted into correct form. It is not easy to determine what the correct form in each case is. Basically, the correct form is the form that has no derivational suffixes. There are, however, also such derived forms that have been lexicalised and have a special meaning.

In (5) we modify the verb stems. There are two stems, *:eleze* and *:angaliwe*, which have the final *e* instead the standard vowel *a*. These are subjunctive forms, and the stem must be converted to standard form, with the final *a*. The stem *:angaliwe* has also a passive marker *-w-*, and it must be removed from the stem.

(5)

```
&ikawa ika :wa V @FMAINV
&tukalazimika tuka :lazimika V @FMAINV
&tukawaeleze tuka +wa :eleza V @FMAINV
&tukayaboresha tuka +ya :boresha V @FMAINV
&akamteua aka +m :teua V @FMAINV
&zikatumika zika :tumika V @FMAINV
&yakawanufaisha yaka +wa :nufaisha V @FMAINV
&likaangaliwe lika :angalia V @FMAINV
&ukatarajia uka :tarajia V @FMAINV
&wakajinunua waka +ji :nunua V @FMAINV
```

Now when we have separated verb components and converted the stems into correct form, we can convert these components into tags. First, we convert the subject prefix and the narrative prefix into tags and move them to the correct place (6).

(6)

```
&ikawa :wa V SUB-PREF=9-SG TAM=NARR:ka @FMAINV
&tukalazimika :lazimika V SUB-PREF=1-PL1 TAM=NARR:ka @FMAINV
&tukawaeleze +wa :eleza V SUB-PREF=1-PL1 TAM=NARR:ka @FMAINV
&tukayaboresha +ya :boresha V SUB-PREF=1-PL1 TAM=NARR:ka
@FMAINV
&akamteua +m :teua V SUB-PREF=1-SG3 TAM=NARR:ka @FMAINV
&zikatumika :tumika V SUB-PREF=10-PL TAM=NARR:ka @FMAINV
&yakawanufaisha +wa :nufaisha V SUB-PREF=6-PL TAM=NARR:ka
@FMAINV
&likaangaliwe :angalia V SUB-PREF=5-SG TAM=NARR:ka @FMAINV
```

```
&ukatarajia :tarajia V SUB-PREF=1-SG2 TAM=NARR:ka @FMAINV
&wakajinunua +ji :nunua V SUB-PREF=1-PL3 TAM=NARR:ka @FMAINV
```

The subject prefixes and narrative prefixes have now been converted into respective tags and moved to the right. The object prefix still needs to be converted into the tag (7).

(7)

```
&ikawa :wa V SUB-PREF=9-SG TAM=NARR:ka @FMAINV
&tukalazimika :lazimika V SUB-PREF=1-PL1 TAM=NARR:ka @FMAINV
&tukawaeleze :eleza V SUB-PREF=1-PL1 TAM=NARR:ka OBJ-
PREF=PL2/PL3 @FMAINV
&tukayaboresha :boresha V SUB-PREF=1-PL1 TAM=NARR:ka OBJ-
PREF=6-PL @FMAINV
&akamteua :teua V SUB-PREF=1-SG3 TAM=NARR:ka OBJ-PREF=SG3
@FMAINV
&zikatumika :tumika V SUB-PREF=10-PL TAM=NARR:ka @FMAINV
&yakawanufaisha :nufaisha V SUB-PREF=6-PL TAM=NARR:ka OBJ-
PREF=PL2/PL3 @FMAINV
&likaangaliwe :angalia V SUB-PREF=5-SG TAM=NARR:ka @FMAINV
&ukatarajia :tarajia V SUB-PREF=1-SG2 TAM=NARR:ka @FMAINV
&wakajinunua :nunua V SUB-PREF=1-PL3 TAM=NARR:ka OBJ-PREF=REFL
@FMAINV
```

Now all verb prefixes have been converted to tags and moved to their respective places. Note that verb prefix tags have been separated by a single space. This ensures that the tags will be kept in the same slot when the strings will be converted to xml-format.

There are still two verbs with subjunctive form that need to be marked with a tag. These are the verbs *tukawaeleze* and *likaangaliwe* (8).

(8)

```
&ikawa :wa V SUB-PREF=9-SG TAM=NARR:ka @FMAINV
&tukalazimika :lazimika V SUB-PREF=1-PL1 TAM=NARR:ka @FMAINV
&tukawaeleze :eleza V SBJN SUB-PREF=1-PL1 TAM=NARR:ka OBJ-
PREF=PL2/PL3 @FMAINV
&tukayaboresha :boresha V SUB-PREF=1-PL1 TAM=NARR:ka OBJ-
PREF=6-PL @FMAINV
&akamteua :teua V SUB-PREF=1-SG3 TAM=NARR:ka OBJ-PREF=SG3
@FMAINV
&zikatumika :tumika V SUB-PREF=10-PL TAM=NARR:ka @FMAINV
&yakawanufaisha :nufaisha V SUB-PREF=6-PL TAM=NARR:ka OBJ-
PREF=PL2/PL3 @FMAINV
&likaangaliwe :angalia V SBJN SUB-PREF=5-SG TAM=NARR:ka
@FMAINV
&ukatarajia :tarajia V SUB-PREF=1-SG2 TAM=NARR:ka @FMAINV
&wakajinunua :nunua V SUB-PREF=1-PL3 TAM=NARR:ka OBJ-PREF=REFL
@FMAINV
```

In the next phase we add English glosses to the stems (9).

(9)

```
&ikawa :wa { be } V SUB-PREF=9-SG TAM=NARR:ka @FMAINV
&tukalazimika :lazimika { be forced } V SUB-PREF=1-PL1
TAM=NARR:ka @FMAINV
&tukawaeleze :eleza { explain } V SBJN SUB-PREF=1-PL1
TAM=NARR:ka OBJ-PREF=PL2/PL3 @FMAINV
&tukayaboresha :boresha { improve } V SUB-PREF=1-PL1
TAM=NARR:ka OBJ-PREF=6-PL @FMAINV
&akamteua :teua { appoint } V SUB-PREF=1-SG3 TAM=NARR:ka OBJ-
PREF=SG3 @FMAINV
&zikatumika :tumika { be used } V SUB-PREF=10-PL TAM=NARR:ka
@FMAINV
&yakawanufaisha :nufaisha { profit } V SUB-PREF=6-PL
TAM=NARR:ka OBJ-PREF=PL2/PL3 @FMAINV
&likaangaliwe :angalia { look at } V SBJN SUB-PREF=5-SG
TAM=NARR:ka @FMAINV
&ukatarajia :tarajia { hope } V SUB-PREF=1-SG2 TAM=NARR:ka
@FMAINV
&wakajinunua :nunua { buy } V SUB-PREF=1-PL3 TAM=NARR:ka OBJ-
PREF=REFL @FMAINV
```

The English tags are surrounded with curly braces. They need to be moved to the right to the appropriate place (10).

(10)

```
&ikawa :wa V SUB-PREF=9-SG TAM=NARR:ka { be } @FMAINV
&tukalazimika :lazimika V SUB-PREF=1-PL1 TAM=NARR:ka { be
forced } @FMAINV
&tukawaeleze :eleza V SBJN SUB-PREF=1-PL1 TAM=NARR:ka OBJ-
PREF=PL2/PL3 { explain } @FMAINV
&tukayaboresha :boresha V SUB-PREF=1-PL1 TAM=NARR:ka OBJ-
PREF=6-PL { improve } @FMAINV
&akamteua :teua V SUB-PREF=1-SG3 TAM=NARR:ka OBJ-PREF=SG3 {
appoint } @FMAINV
&zikatumika :tumika V SUB-PREF=10-PL TAM=NARR:ka { be used }
@FMAINV
&yakawanufaisha :nufaisha V SUB-PREF=6-PL TAM=NARR:ka OBJ-
PREF=PL2/PL3 { profit } @FMAINV
&likaangaliwe :angalia V SBJN SUB-PREF=5-SG TAM=NARR:ka { look
at } @FMAINV
&ukatarajia :tarajia V SUB-PREF=1-SG2 TAM=NARR:ka { hope }
@FMAINV
&wakajinunua :nunua V SUB-PREF=1-PL3 TAM=NARR:ka OBJ-PREF=REFL
{ buy } @FMAINV
```

Now when all tags have been added, we can convert the result into the form, where it should be in the corpus (11).

(11)

```
ikawa wa      V      SUB-PREF=9-SG TAM=NARR:ka      { be }
@FMAINV
tukalazimika    lazimika     V      SUB-PREF=1-PL1 TAM=NARR:ka
{ be forced } @FMAINV
tukawaeleze eleza      V      SBJN SUB-PREF=1-PL1 TAM=NARR:ka
OBJ-PREF=PL2/PL3 { explain } @FMAINV
tukayaboresha boresha      V      SUB-PREF=1-PL1 TAM=NARR:ka
OBJ-PREF=6-PL   { improve } @FMAINV
akamteua     teua     V      SUB-PREF=1-SG3 TAM=NARR:ka OBJ-PREF=SG3
{ appoint } @FMAINV
zikatumika tumika      V      SUB-PREF=10-PL TAM=NARR:ka { be
used } @FMAINV
yakawanufaisha nufaisha     V      SUB-PREF=6-PL TAM=NARR:ka
OBJ-PREF=PL2/PL3 { profit } @FMAINV
likaangaliwe angalia      V      SBJN SUB-PREF=5-SG
TAM=NARR:ka { look at } @FMAINV
ukatarajia tarajia      V      SUB-PREF=1-SG2 TAM=NARR:ka { hope
} @FMAINV
wakajinunua nunua      V      SUB-PREF=1-PL3 TAM=NARR:ka OBJ-
PREF=REFL { buy } @FMAINV
```

The anchors '&' and ':' have been removed and the double spaces have been converted to tabs. This format meets precisely the format of the original corpus.

### 3 The procedure in correcting the corpus

Now when the correction script is ready, we must consider the optimal method of performing the actual corrections. If the corpus would be a single file, it would be easy to take the corpus as input, run the script, and output it as a corrected file. However, the corpus consists of 454 files, and the file names must be retained precisely in the original form. Therefore, we must run each file separately.

This can be done using two alternative methods. In one method, we handle each file separately, run the script, and output it with the same name to another directory. Using this method, we do not destroy the original file. For each file, the same operation is repeated, and the file names are changed accordingly.

In another method, we first prepare a file, where the command for each operation is listed. Using this method, we only need to copy each command at a time and move it to the prompt and run it. Also a macro can be constructed for making the work easier.

If, after correction, it turns out that the correction script needs improvement, the corrections can be made to the script, and the corpus will be run through the corrected script. It is now easier, because the commands saved to the *.bash\_history* file can be copied and used as such. Also here, a macro speeds up the operation.

### 4 Test of the correction script

Below is a piece of the original corpus before and after running the correction script (12).

(12)

Kwa\_hiyo kwa\_hiyo ADV \_ { therefore } @ADVL  
 , , COMMA \_ { , } -  
**tukaangalie** tukaangalie N Heur 9/10-SG { tukaangalie }  
 @<P  
 kama kama ADV \_ { as } @CS  
 hizi hizi PRON DEM 10-PL { these } @<NDEM  
 fedha fedha N 9/10-PL { money } @SUBJ MASS  
 hazitoshi tosha V TAM=NEG-a SUB-PREF=10-PL [tosh] {  
 suffice } @FMAINVtr-OBJ> CAUS SVO VFIN  
 , , COMMA \_ { , }  
 ni ni V -V-BE { are } @FMAINVintr-def  
 vyema vyema ADV \_ { well } @ADVL  
**zikaongezwa** zikaongezwa N Heur 9/10-SG { zikaongezwa }  
 @SUBJ  
 . . \_ { . }  
 Anaweza weza V SUB-PREF=1-SG3 TAM=PR:na [weza] { can }  
 } @FMAINVtr-OBJ> SVO VFIN  
 kuwatuma tuma V NO-TO OBJ-PREF=2-PL3 [tuma] { send }  
 @-FMAINV-n SVO INF  
 wataalam mtaalam N 1/2-PL { expert } @OBJ  
 wake ake PRON POSS 2-PL SG3 { his } @GCON  
**wakamletea** wakamletea N Heur 1/2-PL { wakamletea }  
 @<P  
 pale pale ADV \_ { there } @FMAINVintr-loc LOC-16  
 na na CC \_ { and } @CC  
 kwa\_sababu kwa\_sababu CONJ { because } @CS  
 anawaamini amini V SUB-PREF=1-SG3 TAM=PR:na OBJ-PREF=2-PL3  
 [amini] { believe } @FMAINVtr+OBJ> SVO VFIN  
**wakafanya** wakafanya N Heur 1/2-PL { wakafanya }  
 @OBJ  
 hivyo hivyo ADV \_ { so } @ADVL  
 . . \_ { . }  
 Kwa\_hiyo kwa\_hiyo ADV \_ { therefore } @ADVL  
 , , COMMA \_ { , } -  
 ningeomba omiba V SUB-PREF=1-SG1 TAM=COND:nge [omba] { ask }  
 } @FMAINVtr+OBJ> SVO VFIN  
 hilo hilo PRON DEM 5-SG { this } @ADVL  
**likaangaliwe** likaangaliwe N Heur 9/10-SG {  
 likaangaliwe } @OBJ  
 lisije ja V TAM=SBIN SUB-PREF=5-SG NEG [ja] { come }  
 } @FMAINVintr MONOSLB SV VFIN  
**likawa** likawa N Heur 9/10-SG { likawa } @<P  
 tunasema sema V SUB-PREF=2-PL1 TAM=PR:na [sema] { say }  
 } @FMAINVtr+OBJ> SVO VFIN  
 njaa njaa N 9/10-SG { hunger } @SUBJ  
 hii hii PRON DEM 9-SG { this } @<NDEM  
 ni ni V V-BE NOSUBJ { is } @FMAINVintr-def  
 kwa kwa PREP \_ { for } @ADVL  
 maeneo eneo N 5/6-PL { area } @<P  
 fulani fulani ADJ A-UNINFL { certain } @<NADJ  
 tu tu ADV \_ { only } @FMAINVintr-def

```

na      na      CC      _      { and }      @CC
maeneo   eneo     N      5/6-PL      { area }      @<P
fulani   fulani    ADJ      A-UNINFL      { certain }      @<NADJ
hamna   hamna    V      SUB-PREF=18-SG NEG C:na      { there is not }
          @FMAINVintr VFIN
njaa   njaa     N      9/10-SG      { hunger }      @OBJ
.      .      _      { . }
Kwa_hiyo  kwa_hiyo  ADV      _      { therefore }      @ADVL
,      ,      COMMA      { , }
wasije   ja      V      TAM=SBJN SUB-PREF=2-PL3 NEG [ja]      { come }
}      @FMAINVintr MONOSLB SV VFIN
wakarudi wakarudi  N      Heur 1/2-PL      { wakarudi }
          @SUBJ
hana   hapa     ADV      _      { here }      @FMAINVintr-loc LOC-16
wakasema wakasema  N      Heur 1/2-PL      { wakasema }
          @SUBJ
,      ,      COMMA      { , }
mlisema   sema     V      SUB-PREF=2-PL2 TAM=PAST [sema]      { say }
}      @FMAINVtr-OBJ> SVO VFIN
fanyeni_haraka fanya_haraka  V      IMP [fanya] IMP-PL2
          { hurry }      @FMAINVtr-OBJ> SVO VFIN
,      ,      COMMA      { , }
aah     aah      N      Heur 9/10-SG      { aah }      @OBJ
"<!>" "!"  { ! } <Heur>

```

After running the correction script, the result is as in (13).

```

Kwa_hiyo  kwa_hiyo  ADV      _      { therefore }      @ADVL
,      ,      COMMA      { , }
tukaangalie angalia  V      SBJN SUB-PREF=1-PL1 TAM=NARR:ka
          { look at }      @FMAINV
kama   kama     ADV      _      { as }      @CS
hizi   hizi     PRON     DEM 10-PL { these }      @<NDEM
fedha fedha    N      9/10-PL      { money }      @SUBJ MASS
hazitoshi tosha    V      TAM=NEG-a SUB-PREF=10-PL [tosh]      {
suffice }      @FMAINVtr-OBJ> CAUS SVO VFIN
,      ,      COMMA      { , }
ni      ni      V      V-BE { are }      @FMAINVintr-def
vyema vyema    ADV      _      { well }      @ADVL
zikaongezwa ongeza  V      SUB-PREF=10-PL TAM=NARR:ka      { add }
}      @FMAINV
.      .      _      { . }
Anawenza weza     V      SUB-PREF=1-SG3 TAM=PR:na [weza]      { can }
}      @FMAINVtr-OBJ> SVO VFIN
kuwatuma tuma     V      NO-TO OBJ-PREF=2-PL3 [tuma]      { send }
          @-FMAINV-n SVO INF
wataalam mtaalam   N      1/2-PL      { expert }      @OBJ
wake   ake      PRON     POSS 2-PL SG3 { his }      @GCON
wakamletea leta     V      SUB-PREF=1-PL3 TAM=NARR:ka OBJ-PREF=SG3
          { bring }      @FMAINV
pale   pale     ADV      _      { there }      @FMAINVintr-loc LOC-16

```

na na CC \_ { and } @CC  
 kwa\_sababu kwa\_sababu CONJ { because } @CS  
 anawaamini amini V SUB-PREF=1-SG3 TAM=PR:na OBJ-PREF=2-PL3  
 [amini] { believe } @FMAINVtr+OBJ> SVO VFIN  
**wakafanya** fanya V SUB-PREF=1-PL3 TAM=NARR:ka { do }  
     @FMAINV  
 hivyo hivyo ADV \_ { so } @ADVL  
 . . { . }  
 Kwa\_hiyo kwa\_hiyó ADV \_ { therefore } @ADVL  
 , , COMMA { , }  
 ningeomba omба \_ SUB-PREF=1-SG1 TAM=COND:nge [omba] { ask }  
     @FMAINVtr+OBJ> SVO VFIN  
 hilo hilo PRON DEM 5-SG { this } @ADVL  
**likaangaliwe** angalia V SBJN SUB-PREF=5-SG  
 TAM=NARR:ka { look at } @FMAINV  
 lisije ja V TAM=SBJN SUB-PREF=5-SG NEG [ja] { come }  
     @FMAINVintr MONOSLB SV VFIN  
**likawa** wa V SUB-PREF=5-SG TAM=NARR:ka { be }  
     @FMAINV  
 tunasema sema V SUB-PREF=2-PL1 TAM=PR:na [sema] { say }  
     @FMAINVtr+OBJ> SVO VFIN  
 njaa njaa N 9/10-SG { hunger } @SUBJ  
 hii hii PRON DEM 9-SG { this } @<NDEM  
 ni ni V V-BE NOSUBJ { is } @FMAINVintr-def  
 kwa kwa PREP \_ { for } @ADVL  
 maeneo eneo N 5/6-PL { area } @<P  
 fulani fulani ADJ A-UNINFL { certain } @<NADJ  
 tu tu ADV \_ { only } @FMAINVintr-def  
 na na CC \_ { and } @CC  
 maeneo eneo N 5/6-PL { area } @<P  
 fulani fulani ADJ A-UNINFL { certain } @<NADJ  
 hamna hamna V SUB-PREF=18-SG NEG C:na { there is not }  
     @FMAINVintr VFIN  
 njaa njaa N 9/10-SG { hunger } @OBJ  
 . . { . }  
 Kwa\_hiyo kwa\_hiyó ADV \_ { therefore } @ADVL  
 , , COMMA { , }  
 wasije ja \_ TAM=SBJN SUB-PREF=2-PL3 NEG [ja] { come }  
     @FMAINVintr MONOSLB SV VFIN  
**wakarudi** rudi V SUB-PREF=1-PL3 TAM=NARR:ka { return }  
     @FMAINV  
 hapa hapa ADV \_ { here } @FMAINVintr-loc LOC-16  
**wakasema** sema \_ V SUB-PREF=1-PL3 TAM=NARR:ka { say }  
     @FMAINV  
 , , COMMA { , }  
 mlisema sema \_ V SUB-PREF=2-PL2 TAM=PAST [sema] { say }  
     @FMAINVtr-OBJ> SVO VFIN  
 fanyeni\_haraka fanya\_haraka V IMP [fanya] IMP-PL2  
     { hurry } @FMAINVtr-OBJ> SVO VFIN  
 , , COMMA \_ { , }  
 aah aah N \_ Heur 9/10-SG { aah } @OBJ

"<!>" "!" { ! } <Heur>

All verbs with narrative form were converted to the form, where they should be in the corpus.

## 5 Conclusion

In this report I have shown a method for correcting re-occurring analysis mistakes in an annotated corpus. The correction measures are targeted precisely to the words with faulty analysis, and other words are left intact. The analysis thus achieved may be defective in such cases, where one or more morphemes have more than one interpretation. In such cases all alternatives are presented, whereby the analysis is under-specified. The approach described can be applied to any kinds of corpus correction needs.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/366517866>

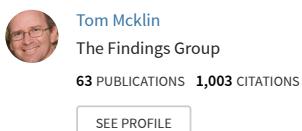
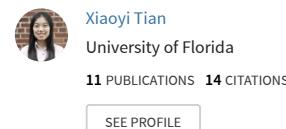
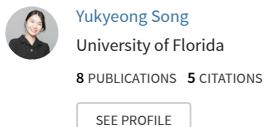
# AI Made By Youth: A Conversational AI Curriculum for Middle School Summer Camps

Conference Paper · December 2022

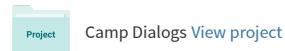
CITATIONS  
0

READS  
221

9 authors, including:



Some of the authors of this publication are also working on these related projects:



# AI Made By Youth: A Conversational AI Curriculum for Middle School Summer Camps

**Yukyeong Song<sup>1</sup>, Gloria Ashiya Katuka<sup>1</sup>, Joanne Barrett<sup>1</sup>, Xiaoyi Tian<sup>1</sup>, Amit Kumar<sup>1</sup>,  
Tom McKlin<sup>2</sup>, Mehmet Celepkolu<sup>1</sup>, Kristy Elizabeth Boyer<sup>1</sup>, Maya Israel<sup>1</sup>**

<sup>1</sup> University of Florida

<sup>2</sup> The Findings Group

{y.song1, gkatuka, jrbarrett, tianx, kumar.amit, mckolu, keboyer, misrael}@ufl.edu, tom@thefindingsgroup.org

## Abstract

As artificial intelligence permeates our lives through various tools and services, there is an increasing need to consider how to teach young learners about AI in a relevant and engaging way. One way to do so is to leverage familiar and pervasive technologies such as conversational AIs. By learning about conversational AIs, learners are introduced to AI concepts such as computers' perception of natural language, the need for training datasets, and the design of AI-human interactions. In this experience report, we describe a summer camp curriculum designed for middle school learners composed of general AI lessons, unplugged activities, conversational AI lessons, and project activities in which the campers develop their own conversational agents. The results show that this summer camp experience fostered significant increases in learners' ability beliefs, willingness to share their learning experience, and intent to persist in AI learning. We conclude with a discussion of how conversational AI can be used as an entry point to K-12 AI education.

## Introduction

Artificial Intelligence (AI) has permeated our lives through technologies such as smart speakers, self-driving cars, and recommendation systems. This technology is not only affecting our daily lives but also changing the future of occupations and job markets (Bughin et al. 2018). Thus, it is imperative to create opportunities for the next generation to learn about the fundamentals of AI and develop positive attitudes towards AI and potential careers in the field. There is an increasing effort to bring AI-related learning experiences to learners at their early ages, with recent studies highlighting the positive effects of these efforts on improving students' knowledge, confidence, and attitudes toward future AI or STEM careers (Wan et al. 2020; Alvarez et al. 2022; Vachovsky et al. 2016).

To engage novice learners in AI learning, we need to consider how to teach AI in relevant and engaging ways. One of the ways to achieve this is to leverage familiar and pervasive technologies such as conversational AIs. Conversational AIs are computer programs with the ability to interact with humans through spoken or textual natural language (Van Brummelen, Heng, and Tabunshchyk 2021). Young

children naturally talk to conversational AIs in their daily lives: children ask Alexa or Google Assistant about their homework or the SAT word of the day and casually express their feelings to them (Garg and Sengupta 2020). By learning about conversational AIs, young learners can be introduced to the basic but fundamental concepts of AI that are addressed in the AI Big Ideas (Touretzky et al. 2019a), such as understanding computers' perception of natural language, the need for training data sets, and AI-human interaction design.

Although there are recent studies on conversational AI curricula and tools for K-12 learners (Van Brummelen, Heng, and Tabunshchyk 2021; Zhu and Van Brummelen 2021), they have primarily focused on online workshop experiences or used an existing interface as a learning tool. This experience report is built upon such previous research and provides a detailed description of a complete conversational AI curriculum utilizing AMBY (AI Made By You), a conversational AI development interface created specifically for our target learners (Kumar et al. 2022). We mapped AI lessons to align with the AI Big Ideas for K-12 learners (Touretzky et al. 2019a) and adopted various pedagogical approaches such as "Use→Modify→Create" (Lee et al. 2011a) and Design Thinking (Thoring, Muller et al. 2011; Arik and Topçu 2020) to design engaging learning activities. We then share results from a series of two-week summer camps in 2021 and 2022. We analyzed campers' pre- and post-survey and video assessments to assess any changes in knowledge and attitudes toward AI. The survey results show that the summer camp experience significantly fostered learners' ability beliefs for, desire to share about, and intent to persist in AI learning. Moreover, students' video assessments demonstrated that they learned to conceptualize AI identifying its core characteristics, such as machine learning. These findings suggest that conversational AIs are one promising entry point for K-12 AI education.

## Related Work

As AI is increasingly integrated into our daily lives, researchers have begun to systematically study how young learners construct understandings of broad AI concepts (Greenwald, Leitner, and Wang 2021) and develop standards of young learners' AI competencies (Kim et al. 2021; Zhou, Van Brummelen, and Lin 2020). In addition, there is

a body of research that shares curricula and tools for teaching AI-related concepts to young learners. For example, Wan et al. (2020) developed SmileyCluster, a collaborative learning environment for teaching machine learning concepts, and showed its positive impacts on students' learning of entry-level machine learning. Similarly, Lin et al. (2020) proposed a chatbot-based curriculum to help young learners understand machine learning concepts. Jordan et al. (2021) built PoseBlocks, a block-based programming environment focusing on helping children understand AI concepts such as face-tracking and emotion recognition. The above studies adopted different AI-related contexts, such as machine learning (Wan et al. 2020; Lin et al. 2020) or face-tracking (Jordan et al. 2021), to introduce AI to K-12 students.

Conversational AIs are computer programs with the ability to interact with humans using natural languages. Conversational AI involves a variety of concepts and knowledge related to AI, such as natural language processing, machine learning, dialogue management, and language generation (Jurafsky and Martin 2021). As the potential benefits of conversational AIs for K-12 AI education have been recognized, researchers have begun to develop tools and curricula to help young children learn about conversational AIs. Zhu and Van Brummelen (2021) developed Convo, a conversational programming agent to teach students about creating conversational agents. Van Brummelen, Heng, and Tabunshchik (2021) developed a curriculum using an existing block-based programming interface called MIT App Inventor to help students build conversational agents integrated into mobile apps.

These studies suggest the promise of conversational AIs in increasing students' interest in AI learning (Zhu and Van Brummelen 2021; Van Brummelen, Heng, and Tabunshchik 2021). This experience report advances knowledge in this space by providing a detailed description of a complete conversational AI curriculum along with its alignment and connections to the AI4K12 Big Ideas (Touretzky et al. 2019a).

## Camp Curriculum

To develop an engaging AI summer camp curriculum for middle school students (rising 7<sup>th</sup> and 8<sup>th</sup> graders), we put together lessons and activities covering four main components: general AI concepts, unplugged activities, conversational AI concepts, and project activities in which the campers develop their own conversational agents. We designed each lesson around the camp's overall and specific learning objectives, described with the phrase "Campers will be able to" in Table 1. These objectives were adapted from AI4K12 progression charts for the 6<sup>th</sup>-8<sup>th</sup> grade band (Touretzky et al. 2019b) and are aligned with the five AI4K12 Big Ideas; 1) Perception: Computers perceive the world using sensors; 2) Representation and Reasoning: Agents maintain representations of the world and use them for reasoning; 3) Learning: Computers can learn from data; 4) Natural Interaction: Intelligent agents require many kinds of knowledge to interact naturally with humans; and 5) Societal Impact: AI can impact society in both positive and negative ways. Table 1 shows the lesson components of the

curriculum mapped to the learning objectives and the corresponding AI4K12 Big Idea. In the following sections, we present detailed descriptions of the four components.

## General AI Lessons

Even though we focused on conversational AI, the broad learning goal of our curriculum is to help learners understand general AI concepts. To achieve that, we designed six lessons to introduce general AI concepts. The contents were extracted from existing open source AI lessons, such as MIT (MIT Raise 2020), Experiments with Google<sup>1</sup>, PBS Learning, and BBC Learning, and adapted to fit a summer camp. The six lessons include: (1) Intro to AI, (2) Intro to Data, (3) Intro to AI and Machine Learning (ML), (4) AI Bias and Ethics, (5) AI Arts, and (6) AI Music. In Intro to AI, campers were introduced to computer science (CS) as "using the power of computers to solve problems," and AI as the branch of computer science that "combined the power of computers" with the "cognitive abilities of humans." Campers also discussed what they recognize as AI around them. In Intro to Data, campers learned how AI needs large amounts of data by interacting with AI applications like Quick Draw (Jongejan et al. 2017) to better understand how AI learns from analyzing many drawings. In Intro to AI and ML, campers were introduced to the relationship between AI and ML to understand how computers learn and how they can teach computers to learn by interacting with tools like ML4kids (Lane 2021) and Teachable Machine (Carney et al. 2020). In AI Bias and Ethics, campers were engaged in discussions about AI bias and ethics around prompts like "what do you think happens to the opinions of people who code AI applications?" In AI & Arts and AI & Music, campers were introduced to applications of AI beyond CS and STEM to Art and Music by examining music and art that had been created through AI experiments.

## Unplugged Activities

Unplugged activities have been employed in various learning contexts to explain CS and AI concepts and improve computational thinking without the use of a computer (Brackmann et al. 2017). Particularly in such environments as summer camps, unplugged activities are valuable for teaching CS and AI concepts and skills without feeling like school. For these reasons, we included some well-known CS unplugged activities such as Human Crane (Code-it 2015), and Sorting Networks (ComputerScienceUnplugged 2010), and we also created some activities, such as the Yoga activity. The Yoga activity is an unplugged activity we created for this camp where students make a "code" by combining a series of yoga poses, such as a "child pose" and the printed-out algorithm blocks, such as "if ... then." After creating the code, they are asked to do the poses following others' code. For each unplugged activity, we explained the purpose of the activity and related them to CS and AI concepts by providing discussion prompts for campers to share their experiences after each activity.

<sup>1</sup><https://experiments.withgoogle.com/>

Components	Lessons & Activities	Learning Objectives “Campers will be able to”	AI4K12 Big Idea
General AI Lessons	1. Introduction to AI	Describe how people sense the environment (e.g., hearing) vs. how computers sense the environment (e.g., using a microphone)	#1: Perception
	2. Introduction to Data	Examine the dataset AI needs to provide meaningful answers	#2: Representation and Reasoning, #3: Learning
	3. Introduction to AI and ML	Describe how data are used for reasoning	#3: Learning
	4. AI Bias and Ethics	Describe ways human biases can be reflected in algorithms	#5: Societal Impact
	5. AI and Arts	Classify images using AI	#1: Perception
	6. AI and Music	Classify sounds using AI	#1: Perception
Conversational AI Lessons	1. Introduction to Chatbots (I)	Describe how a Chatbot functions	#2: Representation and Reasoning
	2. Introduction to Chatbots (II) - (Use) Test Existing Project	Identify the AMBY interface	#2: Representation and Reasoning
	3. Introduction to Intents - (Modify) Special Intents	Identify and create training phrases and responses for special intents	#4: Natural Interaction
	4. Introduction to Intents - (Modify) Existing Intents and (Create) New Intents	Identify and create intents	#4: Natural Interaction
	5. Introduction to Follow-up Intents	Identify and create follow-up intents	#4: Natural Interaction
	6. Conversational Design Principles	Identify conversational design principles and create naturalistic interactions	#4: Natural Interaction
	7. Create a Chatbot from Scratch	Create a chatbot from scratch	#4: Natural Interaction

Table 1: Lessons Mapped to AI4K12 Big Ideas

## Conversational AI Lessons

We provided lessons that were specifically focused on Conversational AIs and chatbots. We created our own conversational AI development environment called AMBY (AI Made By You) to provide a user-friendly experience for creating conversational AIs. The camp’s conversational AI lessons were designed closely around the AMBY interface.

AMBY is designed specifically to support middle school learners in learning about conversational AI and creating their own agents. In the Playground (the main development panel) of AMBY, learners can see, develop, and test their conversational agent (Figure 1). Learners can also deploy their agent on a Google Assistant-compatible device by clicking the “Integrations” button on this page. AMBY offers a list of unique functionalities. First, it allows students to customize the avatar and voice of their agent. Second, it provides a visualization of the conversational flow (as shown in the left panel of Figure 1) with a hierarchical visualization of “Main Intents” and “Follow-up Intents.” Additionally, a testing panel on the right side pane allows users to test their agents through two modalities; typing and voice input. The conversational AI component of the curriculum is composed of seven lessons: (1-2) Intro to Chatbos I & II, (3-4) Intro to Intents I & II, (5) Intro to Follow-up Intents, (6) Con-

versational Design Principles, and (7) Build a Chatbot from Scratch. We used the “Use→Modify→Create” pedagogical approach (Lee et al. 2011a) to provide scaffolding during guided hands-on lessons. We provided three sample projects for campers to use and a template project for them to modify through several lessons (Figure 2). Most of the lessons were designed with a short content and longer hands-on practice, tailored towards novice programmers. In the following sections, we describe each lesson in detail.

**Lesson 1: Intro to Chatbots (I)** This lesson provides an introduction to conversational apps or chatbots<sup>2</sup> in an interactive and relatable way. It begins with warm-up questions such as “Have you ever asked your smartphone a question?” and “Did it give you the answer you wanted?”. Then, through a demo interaction with a Google Home Mini device, campers asked questions and gauged the Google Assistant’s responses. They were introduced to three main kinds of chatbots, namely Rule-based, Intelligent, and AI-based chatbots, and the various things chatbots can do, such as answering questions and making recommendations. Then, they brainstormed what chatbots they would like to create and

<sup>2</sup>During the camp, we used the term chatbots to refer to conversational apps for ease of understanding.

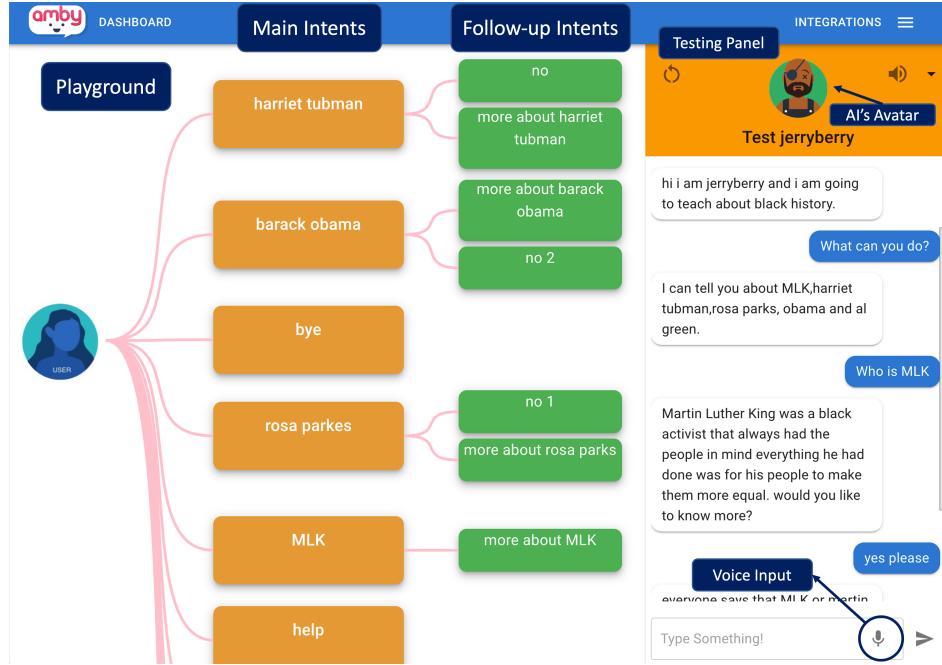


Figure 1: AMBY development environment. The screenshot is the development page for a student project, JerryBerry

why users would need them.

**Lesson 2: Intro to Chatbots (II)** This lesson aims to introduce the idea of becoming developers of their own conversational apps and working within AMBY. As a warm-up activity, campers were introduced to using a stack of customized cards with samples of developer goals, user utterances, and chatbot responses. Campers were encouraged to group the cards based on chatbot ideas. For instance, one group of cards might contain a developer goal: “a chatbot that recommends music”; user’s utterance: “Can you please recommend a fun song?” and the chatbot’s response: “Sure, Wobble by V.I.C is a fun song.” This activity applied the fundamental conversational AI concepts of intents, training phrases, and responses. Then, campers were introduced to AMBY and interacted with pre-built chatbots in the system.

**Lesson 3: Intro to Intents (I) - Special Intents** This lesson covers the concept of intents and special intents. In AMBY, intents represent a state of conversation defined by the user’s goal. Developers train the chatbot to recognize a given intent from user input using a set of “training phrases” (example inputs) and designate a list of “responses” for the chatbot to return. For hands-on practice, campers modified the template chatbot “AboutMeBot” across several lessons (see Figure 2). In Lesson 3, campers modified the *special* intents, which are the default intents that the system creates for every agent. Campers customized the “Greet” intent, which is meant to exchange greetings between a user and the chatbot, and the “Default Fallback” intent, specifying what the chatbot will utter when it does not understand what the user says. For example, campers can change the generic Greet intent responses like, “Hi, how are you today” to something

more relevant, like “Hi, I’m AboutMeBot. You can ask me questions about the person who developed me.”

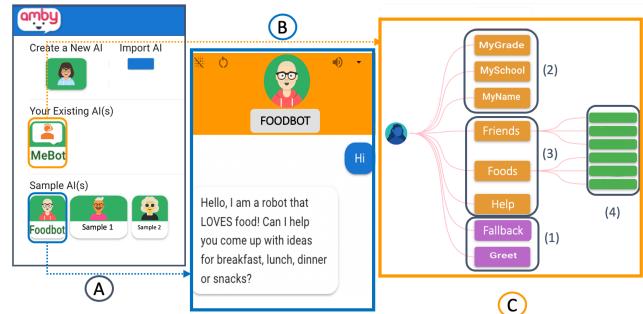


Figure 2: Progression of Conversational AI Lessons within the interface: (A) Dashboard page with sample AIs for campers to use, and existing AI, AboutMeBot, to modify; (B) Testing panel for campers to interact with chatbots; (C) Playground page for AboutMeBot with (1) Special intents, (2) Existing intents, (3) New intents and (4) Follow-up intents created by campers during lessons

**Lesson 4: Intro to Intents (II) - Modify Existing Intents and Create New Intents** This lesson is mostly hands-on, with campers further modifying the “AboutMeBot”. Campers added more training phrases, created new intents, and updated the existing responses to personalize their chatbot. For the “AboutMeBot”, campers created new intents, such as “MyFavoritefood”, and added training phrases and responses.

**Lesson 5: Intro to Follow-up Intents** This lesson introduces the concept of follow-up intents, which are in-

tents that are linked to another intent. When the original intent is matched, the next thing the user says is matched to the follow-up intent. Campers created a new intent for the “AboutMeBot” called “FavoriteFood” and added three follow-up intents, such as “FavoriteBreakfast”, “FavoriteLunch”, “FavoriteDinner.”

**Lesson 6: Conversational Design Principles** This lesson focuses on five conversational design principles: setting user’s expectations, conversational flow, conversational markers, “No match” errors, and “Help” responses. Campers modified the responses for the “AboutMeBot” to incorporate these conversational design principles. For example, they modified the “Greet” intent response from “Hi, I’m AboutMeBot, You can ask me questions about the person who developed me” → “Hi, I’m AboutMeBot, You can ask me questions about the *name, grade, school, favorite food, and favorite color* of my developer.” Additionally, campers were taught how to customize the chatbot’s voice and the avatar that represents their chatbot.

**Lesson 7: Build a Chatbot from Scratch** This lesson provides an end-to-end hands-on experience in creating a chatbot. Campers were placed in groups and provided with a worksheet to guide them. There was minimal involvement from camp facilitators, who monitored progress through the checklist on the worksheet and offered help when asked by the campers. This lesson helped campers gain the confidence to develop their own chatbots more independently.

### Project Activities

After learning about conversational AI through lessons, students engaged in the project activities, where they developed their own conversational agents. To guide their projects, we provided a Project Design Log, developed based on Design Thinking (Arik and Topçu 2020). We chose Design Thinking because of its focus on empathy, which helped students think about the social impact of AI (Big Idea #5).

Students began with an individual project and proceeded to a collaborative, pair-programming project, in which a pair of campers switched roles between the *driver* and the *navigator*. The *driver* is responsible for creating intents and typing training phrases and responses, while the *navigator* provides support by coming up with the training phrases and checking errors (Celepkolu and Boyer 2018). For pair programming, campers were asked to individually brainstorm 3-5 project ideas and were paired based on these ideas. Both project activities followed the Design Thinking (Thoring, Muller et al. 2011; Arik and Topçu 2020) process: Empathize, Define, Ideate, Prototype, Test.

## Implementation and Outcomes

### Camp Context

**1) Year 1 (2021)** In the summer of 2021, we conducted a two-week-long day camp as the first implementation with 14 rising seventh and eighth graders (2 girls and 12 boys, 11 Black/African-American, and 3 White students). The average age of campers was 12.3 (SD=1). Ahead of the camp,

we held a one-week professional development to prepare our undergraduate facilitators. In the first year, campers developed their own conversational agents using Google Dialogflow, which is a conversational AI development tool more suited to adult users. The campers in year 1 experienced frustration working with Dialogflow because of the dense text displayed in the interface and difficulty with typing. All these contributed to the team moving forward with the development of AMBY.

**2) Year 2 (2022)** Two camp sessions were conducted the following year with two major changes to the curriculum. The first was to incorporate the AI4K12 Big Ideas Guidelines aligned with the released curriculum Progressions (Touretzky et al. 2019a). The second was the development team’s effort to design AMBY as an interface to make learner interaction smoother and more accessible.

We developed AMBY following four iterative development cycles, beginning with a formative user study in the context of summer camp 2021 and followed by two rounds of usability studies. The findings from these iterative user studies have led us to create the fully functional prototype of AMBY before the year 2 camps. New lessons were created to address general AI in accordance with the AI Big Ideas, as well as to teach learners about the new interface.

Overall, 32 campers participated in the year 2 camps (17 girls and 15 boys, 25 Black/African-American, 5 Hispanic/Latinx, 4 White, 1 Asian, 1 Native American/Alaskan Native)<sup>3</sup>. The average age was 12.7 (SD=0.7). Before the camp started, eight undergraduate facilitators participated in a three-week professional development, extended from the previous year’s single week. In the professional development, we addressed facilitators’ roles, provided facilitators with opportunities to learn about AI and conversational AI, and invited them to practice teaching lessons in the form of micro-teaching. They also created conversational agents using AMBY. Between the two sessions, we held a three-day-long mini-professional development to address changes in the lessons, activities, and schedule.

### Detailed Schedule

We organized the camp curriculum into a camp schedule that guided the daily flow of activities for the camp. Table 2 shows our week 1 camp schedule.

### Outcomes

#### 1) AI Attitude Survey

We gathered pre/post-survey data from 32 campers who participated in the second year along four constructs: ability beliefs, sharing, identity, and persistence. The ability beliefs construct drew items from the BASICS-SQ (Outlier Research & Evaluation 2017) and focused on perceptions of students’ ability to understand AI, with items like: “I can do well in AI” and “I can figure out how to solve hard AI problems if I try.” Sharing is from the Personal Creativity Scale (McKlin et al. 2018) and asks students to report agreement/disagreement with prompts like: “I want to share what

<sup>3</sup>Four campers identified as more than one race/ethnicity.

Time	Monday	Tuesday	Wednesday	Thursday	Friday
10:30	Opening Event	Intro to Chatbots (I)	Intro to Data	Yoga Activity	Individual Project Development
11:10	Ice Breakers	Minefield	Intro to AI & ML	Design Thinking	Peer Testing & Feedback
11:45	Pre-survey	AI Ethics & Bias	Intro to Intents (II) - Create New Intents	Individual Project - Empathize, Define	Peer Testing & Feedback
12:20			Lunch		
13:00	Pre-assessment	Intro to Chatbots (II)	Intro to Follow-up Intents	Individual Project - Ideate	AI & Arts
13:35	Pre-video	Facilitators Project Showcase - Use Existing Projects	Conversational Design Principles	Individual Project - Prototype	Finalize Individual Projects
14:25	Intro to AI	Intro to Intents (I) - Modify Special Intents	Modify Existing Responses	Individual Project Development	Interview on AMBY
15:00	Human Crane	Musical Dots	Create a Chatbot from Scratch	Careers in STEM	Fun Friday
15:35	Wind down/Gym time	Wind down/Gym time	Wind down/Gym time	Wind down/Gym time	Fun Friday

Table 2: Camp Schedule - Week 1

I do in the camp with my friends.” The identity construct asks students whether they perceive that they have options in AI/STEM careers and is adapted from the BASICS-SQ with prompts like: “I see myself using AI in my future job.” Persistence is also adapted from the BASICS-SQ future time perspective construct and is distinct from identity in that it focuses on actions students might take in the near future related to AI learning. The prompts include “I would like to join an AI club” and “I would like to learn more about AI in the future.” We used campers’ composite scores from each construct to conduct a paired-measures t-test comparing pre and post-responses. Table 3 shows significant increases from pre-to-post on three of the four constructs: ability beliefs, sharing, and persistence<sup>4</sup>.

Construct	Mean	p	Effect size
<i>Ability Beliefs</i> (n = 31)	Pre 2.91 Post 3.30	0.006**	0.530
<i>Sharing</i> (n = 32)	Pre 2.90 Post 3.18	0.049*	0.362
<i>Identity</i> (n = 31)	Pre 2.84 Post 3.00	0.275	0.200
<i>Persistence</i> (n = 31)	Pre 2.77 Post 3.10	0.019*	0.447

Note. \* p <.05; \*\* p <.01; Effect size calculated using Cohen’s D

Table 3: Pre/Post Comparison by Attitudinal Constructs

## 2) Video Assessment

To identify campers’ learning about AI, we analyzed video recordings in which campers were tasked to answer, “What is AI?” as if they were explaining AI to their family or friends. They recorded short videos both before and after the camp. In both pre and post-videos, campers conceptualized AI with such key words as “robot”, “made by human (artificial)”, “smart (intelligent)”, and “assisting/helping.” Also, they often mentioned that AI “talks back to you/can have a conversation with you.” Many of them gave a list of what AI can do, such as “telling you a joke” or “helping you with homework”. They also gave examples of in-service AIs or

<sup>4</sup>Differences in N in the table for each construct is attributable to the fact that one participant skipped some items.

IoT (Internet of Things) products, such as “Smart TVs”, “refrigerators”, or smart speakers such as “Siri”, “Alexa”, and “Google Home”. In the pre-videos, we found more campers showed uncertainty about their knowledge of AI. For instance, three out of 32 campers only said “I don’t know” in the pre-videos, but they provided better answers in the post-videos, including keywords like “smart”, “data”, “chatbot”, and “self-driving cars”.

## 3) Students’ Projects

Learners created 58 conversational AI projects utilizing AMBY. Project topics varied depending on their interests, including game/sports tutorials, music/movie recommendations, joke telling, information giving, and mental health. We present two examples of projects from the 2022 camp sessions.

1. **Jerry Berry** is a conversational agent that gives information about Black history and influential Black figures, such as Martin Luther King Jr. and Barack Obama. This project was built collaboratively by two African-American students. While developing this project, they utilized effective conversational design principles that were taught in the conversational AI lessons (Lesson 6) to make the conversation more natural. For example, to avoid monologue in the agent’s responses and a better conversational flow, they broke down the description of Black influencers into nested intents in which the agent’s response ends with a question “Would you like to know more?” so that users can choose whether to continue the conversation or not.

2. **ZooBot** introduces interesting facts about animals, as well as tips for people to defend themselves against dangerous animals. ZooBot was one of the projects with the most intents (10 main intents, 38 follow-up intents, and 22 follow-ups of follow-up intents). In the project demo scripts, one of the students shared their mental models of the conversational flow of their agent:

*We added many different training phrases so our agent can easily understand what the user is attempting to ask. The following intent we have is the ‘greet’ intent, which is how the bot asks the user the question on how they would like to proceed. The next intent is the ‘yes’ intent. It recognizes when someone wants to receive animal facts. The ‘animal’ intent can be trig-*

*gered after the ‘yes’ intent, and it will say what animals it can provide facts about.*

The demonstration of their project showed a clear understanding of AI, derived from our lessons. For example, they understood the role of training data, which can make their agent more likely to understand the user’s intent (taught in lesson 3 and 4); they demonstrated the importance of natural AI-human interactions (lesson 6) by authoring multiple follow-up intents (lesson 5) and allowing users to take multiple conversational turns.

## Discussion

In this paper, we shared an innovative conversational AI curriculum for middle school summer camps. The design of the curriculum is closely connected to AI4K12 Big Ideas (Touretzky et al. 2019a) and the development team’s work on the AMBY interface. Findings suggest that this approach successfully supported middle school learners in gaining a well-rounded understanding of AI. Some components of our curriculum addressed curricular suggestions from previous studies. For example, the content of the conversational AI lessons aligned with the “app-building tutorials” and “Alexa skill tutorials” of Van Brummelen, Heng, and Tabunshchiky (2021)’s work. In addition, our curriculum provided more support and scaffolding by adopting Use→Modify→Create pedagogical approach (Lee et al. 2011b) and Design Thinking (Thoring, Muller et al. 2011). The biggest difference in the current work is the co-design and integration between our curriculum and our conversational AI development interface, AMBY. While Van Brummelen, Heng, and Tabunshchiky (2021) used an existing tool, MIT App Inventor, we created our interface specifically for middle school learners. AMBY and the camp curriculum were developed together in a synergistic manner: we included lessons, tutorials, and hands-on activities specifically to help campers understand and use AMBY. At the same time, when we observed children encountering usability challenges with AMBY during the camp, our development team was able to make formative changes to improve the interface.

The implementation of our curriculum suggests the promising potential of conversational AI as an effective entry point to K-12 AI education in informal settings. We found a significant increase in campers’ attitudes in three out of four constructs, including ability beliefs, sharing, and persistence in AI learning. These findings indicate improvements in students’ beliefs in their ability to understand AI and to create projects using AI, in their comfort with sharing their work with friends and family, and in actions related to AI and learning AI that they would like to take in the future. There was no significant difference in the identity construct. One possible reason to explain this null result is that we intentionally did not describe AI-related jobs as the “best” career path. Since most of our campers have not started or are in the early stage of forming professional identities (Erikson 1993), we wanted to give these youth time to explore various possibilities before deciding on their future careers.

We have also reported findings from the pre-post videos where campers explain what AI is. It is notable that more

campers tried to conceptualize AI by mentioning the characteristics of AI in the post-videos instead of merely listing the examples of AI services, which was seen more often in the pre-videos. For example, in one post-video, a student said, “AI learns and becomes intelligent based on the output you give, and then it shows intelligence through inputs you give it as tasks”. This suggests that the student has begun to grasp the basic concepts of training and test data and how they are used for machine learning, which is related to the AI Big Idea number 3: “Computers can learn from data” (Touretzky et al. 2019a).

Lastly, from the projects created by campers, we found that our lessons and project activities helped them learn about important AI Big Ideas, such as “Natural Interaction” (Big Idea 4), and the “Social Impacts of AI” (Big Idea 5). In conversational AI lessons, we provided a series of conversational design principles for natural human-AI interaction design (lesson 6). We could observe campers applying these principles by adding greetings and social talk or breaking agents’ information-giving monologues into nested intents (see Jerry Berry example). This suggests campers’ learning gain in AI Big Idea #4: “Making agents interact comfortably with humans is a substantial challenge for AI developers” (Touretzky et al. 2019a). In addition, when generating project ideas, students started by empathizing with people around them, a step in line with Design Thinking. For the Jerry Berry project, for example, the campers considered the social impact of their agent that tells stories of successful Black figures in history to empower Black people, which is related to the AI Big Idea #5: “AI applications can impact society in both positive and negative ways” (Touretzky et al. 2019a).

## Conclusion

We have reported on a novel conversational AI curriculum for middle school summer camps. Our curriculum consists of four main components: general AI lessons, unplugged activities, conversational AI lessons, and project activities. Each component is composed of a series of lessons and hands-on activities. The outcomes of the implementation indicate the promise of conversational AI as an entry to K-12 AI education, identifying the positive impacts on students’ attitudes, AI conceptualization, and understanding of the social impacts of AI. The findings suggest many important directions for future work. First, there is much to explore regarding how to foster general AI knowledge through the specific context of conversational AI. Second, this work has been conducted in an informal setting, which limited the opportunity to formally assess learner knowledge. Future work should investigate the assessment of learning in the context of conversational AI. Finally, future work should investigate conversational AI learning experiences with a broader set of learners in terms of geography, race/ethnicity, and age.

## Acknowledgements

This material is based upon work supported by the National Science Foundation under grant DRL-2048480. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

- Alvarez, L.; Gransbury, I.; Cateté, V.; Barnes, T.; Ledéczi, ; and Grover, S. 2022. A Socially Relevant Focused AI Curriculum Designed for Female High School Students. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 12698–12705.
- Arik, M.; and Topcu, M. S. 2020. Implementation of engineering design process in the K-12 science classrooms: Trends and issues. *Research in Science Education*, 1–23.
- Brackmann, C. P.; Román-González, M.; Robles, G.; Moreno-León, J.; Casali, A.; and Barone, D. 2017. Development of computational thinking skills through unplugged activities in primary school. In *WiPSCE '17: Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 65–72.
- Bughin, J.; Hazan, E.; Lund, S.; Dahlström, P.; Wiesinger, A.; and Subramaniam, A. 2018. Skill shift: Automation and the future of the workforce. *McKinsey Global Institute*, 1: 3–84.
- Carney, M.; Webster, B.; Alvarado, I.; Phillips, K.; Howell, N.; Griffith, J.; Jongejan, J.; Pitaru, A.; and Chen, A. 2020. Teachable machine: Approachable Web-based tool for exploring machine learning classification. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems*, 1–8.
- Celepkolu, M.; and Boyer, K. E. 2018. The importance of producing shared code through pair programming. In *Proceedings of the 49th ACM technical symposium on computer science education*, 765–770.
- Code-it. 2015. Human crane: Code-it supported by HIAS, Hampshire Inspection and Advisory Service. <https://code-it.co.uk/ks1/crane/humancrane>. Accessed: 2022-12-12.
- ComputerScienceUnplugged. 2010. Classic CS Unplugged. <https://classic.csunplugged.org/documents/activities/sorting-network/unplugged-08-sorting-networks-2010.pdf>. Accessed: 2022-12-12.
- Erikson, E. H. 1993. *Childhood and society*. WW Norton & Company.
- Garg, R.; and Sengupta, S. 2020. Conversational technologies for in-home learning: using co-design to understand children's and parents' perspectives. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–13.
- Greenwald, E.; Leitner, M.; and Wang, N. 2021. Learning Artificial Intelligence: Insights into How Youth Encounter and Build Understanding of AI Concepts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 15526–15533.
- Jongejan, J.; Rowley, H.; Kawashima, T.; Kim, J.; and Fox-Gieg, N. 2017. Google Quick, Draw. <https://quickdraw.withgoogle.com/>. Accessed: 2022-12-12.
- Jordan, B.; Devasia, N.; Hong, J.; Williams, R.; and Breazeal, C. 2021. PoseBlocks: A toolkit for creating (and dancing) with AI. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 15551–15559.
- Jurafsky, D.; and Martin, J. H. 2021. Chapter 24: Chatbots Dialogue Systems. In *Speech and Language Processing*.
- Kim, S.; Jang, Y.; Kim, W.; Choi, S.; Jung, H.; Kim, S.; and Kim, H. 2021. Why and what to teach: AI curriculum for elementary school. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 15569–15576.
- Kumar, A.; Tian, X.; Celepkolu, M.; Israel, M.; and Boyer, K. E. 2022. Early Design of a Conversational AI Development Platform for Middle Schoolers. In *2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 1–3. IEEE Computer Society.
- Lane, D. 2021. *Machine learning for kids*. No Starch Press.
- Lee, I.; Martin, F.; Denner, J.; Coulter, B.; Allan, W.; Erickson, J.; Malyn-Smith, J.; and Werner, L. 2011a. Computational thinking for youth in practice. *ACM Inroads*, 2(1): 32–37.
- Lee, I.; Martin, F.; Denner, J.; Coulter, B.; Allan, W.; Erickson, J.; Malyn-Smith, J.; and Werner, L. 2011b. Computational Thinking for Youth in Practice. *ACM Inroads*, 2(1): 32–37.
- Lin, P.; Van Brummelen, J.; Lukin, G.; Williams, R.; and Breazeal, C. 2020. Zhorai: Designing a conversational agent for children to explore machine learning concepts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 13381–13388.
- McKlin, T.; Magerko, B.; Lee, T.; Wanzer, D.; Edwards, D.; and Freeman, J. 2018. Authenticity and personal creativity: How EarSketch affects student persistence. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 987–992.
- MIT Raise. 2020. MIT AI Literacy Units. <https://raise.mit.edu/resources.html>. Accessed: 2022-12-12.
- Outlier Research & Evaluation. 2017. BASICS Study ECS Student Implementation and Contextual Factor Questionnaire Measures [Measurement scales]. Technical report, Outlier Research & Evaluation at UChicago STEM Education | University of Chicago, Chicago, IL.
- Thoring, K.; Muller, R. M.; et al. 2011. Understanding design thinking: A process model based on method engineering. In *DS '69: Proceedings of E&PDE 2011, the 13th International Conference on Engineering and Product Design Education, London, UK*, 493–498.
- Touretzky, D.; Gardner-McCune, C.; Martin, F.; and Seehorn, D. 2019a. Envisioning AI for K-12: What should every child know about AI? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9795–9799.
- Touretzky, D. S.; Gardner-McCune, C.; Martin, F.; and Seehorn, D. 2019b. K-12 guidelines for artificial intelligence:

what students should know. Presented at ISTE19, the 2019 Conference of the International Society for Technology in Education.

Vachovsky, M. E.; Wu, G.; Chaturapruek, S.; Russakovsky, O.; Sommer, R.; and Fei-Fei, L. 2016. Toward more gender diversity in CS through an artificial intelligence summer program for high school girls. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 303–308.

Van Brummelen, J.; Heng, T.; and Tabunshchyk, V. 2021. Teaching tech to talk: K-12 conversational artificial intelligence literacy curriculum and development tools. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 15655–15663.

Wan, X.; Zhou, X.; Ye, Z.; Mortensen, C. K.; and Bai, Z. 2020. SmileyCluster: Supporting Accessible Machine Learning in K-12 Scientific Discovery. In *Proceedings of the Interaction Design and Children Conference*, IDC ’20, 23–35. New York, NY, USA: Association for Computing Machinery. ISBN 9781450379816.

Zhou, X.; Van Brummelen, J.; and Lin, P. 2020. Designing AI learning experiences for K-12: emerging works, future opportunities and a design framework. *arXiv preprint arXiv:2009.10228*.

Zhu, J.; and Van Brummelen, J. 2021. Teaching students about conversational AI using Convo, a conversational programming agent. In *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 1–5. IEEE.