Modeling Knowledge and Functional Intent for Context-Aware Pragmatic Analysis

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Nikhita Vedula, B.Tech, MS

Graduate Program in Computer Science and Engineering

The Ohio State University

2020

Dissertation Committee:

Dr. Srinivasan Parthasarathy, Advisor

Dr. Huan Sun

Dr. Eric Fosler-Lussier

Dr. Duane Wegener

Abstract

The advent and proliferation of web technologies and the boom of big data has given rise to new modes of social interactions, as well as a deluge of information across numerous domains, topics and languages. This unstructured human-generated data contains rich semantic and stylistic signals reflecting the latent intentions of people, and is remarkably valuable for knowledge discovery. In order to effectively understand and lend structure to such massive quantities of data, it is not enough to merely extract and analyze salient patterns from it. It is equally important to understand and model the functional intentions, behavioral characteristics, reactions and responses of the *authors* and/or *consumers* of the data in question. In this dissertation, we extract and study the functional meaning, intentions and knowledge patterns in modern digital content in disparate contexts, which is the focus of an area called *latent pragmatic analysis*. We then propose some interesting avenues of future research.

Our first contribution is the development of context-aware knowledge harvesting techniques to automatically organize unstructured information into easily accessible, hierarchical schemas, thus avoiding the need for manual or expert curation. We propose a technique, `ETF`, that learns a ranking model utilizing semantic and graph theoretic features to insert newly emerging conceptual information into large, existing general-purpose knowledge stores like DBPedia. Second, we develop a machine learning algorithm, `BOLT-K`, to automatically learn ontology hierarchies for emergent topics or sub-domains. It significantly reduces the need for

human supervision by augmenting the limited labeled training data, and transferring knowledge from existing, functionally related schemas. Third, to ensure the presence of factually accurate information in our constructed knowledge stores, we propose an encoder-decoder framework called FACE-KEG. It predicts the veracity of input textual information via a graph transformer network classifier, and also generates human-comprehensible explanations via a recurrent neural network, to justify its predictions.

Orthogonal to our prior efforts of studying and structuring digital content from the web, our subsequent work transitions into modeling the behavioral aspects and intentions of human users or creators of digital content. As our fourth and fifth contributions, we discovered and categorized user intentions or *intents* (our proposed technique is called OPINE), and *domains* (our proposed technique is called ADVIN) from human user conversations and user interactions with virtual assistants. Our deep learning models are independent of the topic of the input text utterances, require minimal domain-specific labeled training data and can be employed for various downstream conversational and search applications. Our final contributions augment the textual semantics and language cues with emotional signals, social network topology and alternate modalities (e.g. video, audio), to investigate multiple facets of online user behavior towards social good applications. To this end, our sixth contribution analyzed the multimodal dependency patterns between digital multimedia content attributes and the responses and reactions invoked among their target users or viewers. Our seventh contribution is a scalable, unsupervised model to realize the credibility and reliability of online social network users, by quantifying appropriate socio-psychological elements such as the social influence exerted by users in a network, the underlying network structure and the affective valence expressed in user content. Our eighth contribution successfully correlates distinctive *online* social platform cues (e.g. user activity, network engagement,

affective valence, linguistic patterns) with theoretical insights from *offline* medical and psychological findings, with emphasis on clinical depression.

Through these research efforts, we emphasize that developing computational models enriched by domain specific insights and contextual information that adapt to users' behavior, needs and operations under various settings is of immense value in several disciplines, such as information retrieval, conversational search, marketing and crisis response. The long-term research goal of this dissertation is to develop novel, intelligent, context-aware systems that effectively unify and structure heterogeneous data from trustworthy sources and respond to human intents and behavior patterns.

Dedicated to my parents, and all associated with The Ohio State University.

Acknowledgments

At the very outset, I would like to express my heartfelt gratitude for my advisor Prof. Srinivasan Parthasarathy (Srini). He is one of the nicest and kindest people that I have ever met. Right from the beginning, Prof. Srini has always given me the freedom to discover and explore my research interests, come up with my own problem statements, and work at my own pace. I have always admired and have been inspired by his dedication and passion towards research and the progress of his students. For example, even during weekends and late at night, he was always available to give detailed feedback on my research projects, scientific papers as well as conference presentations. His constructive and insightful advice has been invaluable whenever I was stuck at any point in my research. He has always encouraged me to explore a particular problem with respect to new domains or dimensions, to understand the bigger picture, and have a clear rationale behind every step of our solutions to the problems we solve. Over the past five years, Prof. Srini not only encouraged me to do good research, but was also instrumental in my overall growth and development as a computer scientist. He made sure I gained useful experience in crucial academic activities such as scientific grant and proposal writing, reviewing conference and journal papers, giving guest lectures and mentoring students. He always recommended me for, and motivated me to apply for various grants and fellowships throughout my PhD. He supported my participation in multiple events to improve the representation of women in computer science. He also encouraged us to attend different conferences, despite having only workshop or demo papers

(or sometimes even no papers) accepted there. Multiple times, he displayed more confidence than me in my capabilities and success as a researcher. He spent a notable amount of time and effort helping me with my internship search and job search, by connecting me to relevant peers in the industry and academia and providing useful advice on my application documents. I learnt indispensable life lessons during my time with him as a PhD student at Ohio State, which will serve me not only throughout my career, but also my whole life. Without a doubt, I could not have completed my Ph.D. without the motivation, support and guidance of Prof. Srini.

I would also like to thank all my committee members, Dr. Huan Sun, Dr. Eric Fosler-Lussier, Dr. Duane Wegener and Dr. Kannan Srinivasan (Ph.D. candidacy committee). They provided me valuable feedback, advice and ideas, and asked insightful questions during my candidacy and PhD defense that helped me develop some of my current and future work, and improve the quality of this dissertation. The work done during my PhD has been supported at various stages by the National Science Foundation under grant EAR-1520870, a Graduate School Fellowship and Presidential Fellowship from the Ohio State University, and gift funding from Adobe Research. Computational resources have been provided by the Ohio Supercomputer Center (grants PAS0166 and PAA0205) and the Department of Computer Science and Engineering at OSU. All opinions, findings, conclusions and recommendations expressed in this dissertation material represent the opinion of the author and her collaborators, which are not necessarily shared or endorsed by their sponsors.

I have been fortunate to work with and be mentored by excellent researchers outside of OSU, both as part of my summer internships and research projects. I would like to express my sincere gratitude to all my internship mentors: Dr. Alessandra Sala, Dr. Patrick K. Nicholson, Dr. Deepak Ajwani and Dr. Sourav Dutta at Nokia Bell Labs Ireland; Dr.

Nedim Lipka at Adobe Research; and Dr. Rahul Gupta, Aman Alok, Mukund Sridhar and Dr. Shankar Ananthakrishnan at Amazon Alexa AI. They have played a crucial role in mentoring and supporting me both during and after my internships, and collaborations with them have resulted in successful research projects that are all part of my PhD dissertation. I would further like to express my thanks to my close collaborators for their insightful suggestions, useful discussions, valuable ideas and help with writing scientific papers: Dr. Valerie Shalin, Pranav Maneriker, Dr. Krishnaprasad Thirunarayan, Dr. Munira Syed, Wei Sun, Harsh Gupta, Dr. David Fuhry, Dr. Jiayong Liang, Dr. Hemant Purohit, Dr. Mitsunori Ogihara, Dr. Joseph Johnson, Dr. Gang Ren, Hyunhwan Lee, Saravana Kumar, Dr. Omar El-Khoury, Dr. Amit Sheth, Dr. Desheng Liu, Dylan Wood, Dr. Ethan Kubatko, Molly Moran and Dr. Hussein Al-Olimat.

I would next like to convey my thanks to all my peers at the Data Mining Research Laboratory at OSU: Dr. Aniket Chakrabarti, Bortik Bandyopadhyay, Goonmeet Bajaj, Dr. Jiankai Sun, Dr. Jiongqian Liang, Meghana Moorthy Bhat, Moniba Keymanesh, Saket Gurukar, Saumya Sahai, Vedang Patel, Dr. Yu Wang and Yuntian He. Fruitful discussions with many of them have given birth to successful research ideas throughout my PhD. Their constructive suggestions also helped me improve the quality of my research papers, talks and presentations.

Finally, I owe my deepest and most sincere gratitude to my parents for their unconditional love, support, advice and encouragement throughout the course of my PhD. They have never failed to celebrate my successes, and motivate me to persevere after my failures. Without them, I am nothing.

Vita

July 2011 – May 2015	B.Tech, Department of Computer Science and Engineering, Visvesvaraya National Institute of Technology, Nagpur, India.
August 2015 – present	Ph.D. Student, Department of Computer Science and Engineering, The Ohio State University, USA.
August 2015 – August 2016	University Graduate Fellow, The Ohio State University, USA.
August 2016 – December 2019	Graduate Research Associate, Department of Computer Science and Engineering, The Ohio State University, USA.
May 2017 – August 2017	Data Analytics Research Intern, Nokia Bell Laboratories, Ireland.
May 2018 – August 2018	Data Science Research Intern, Adobe Research, USA.
May 2019	MS, Department of Computer Science and Engineering, The Ohio State University, USA.
May 2019 – August 2019	Applied Scientist Intern, Amazon Alexa AI, USA.
January 2020 – present	Presidential Fellow, The Ohio State University, USA.

Publications

Nikhita Vedula, Rahul Gupta, Aman Alok and Mukund Sridhar. "Automatic Discovery of Novel Intents & Domains from Text Utterances." In *arXiv preprint arXiv:2006.01208*, 2020.

Nedim Lipka and **Nikhita Vedula**. "Utilizing recurrent neural networks to recognize and extract open intent from text inputs." In *US Patent App. 16/216,296*, 2020.

Nikhita Vedula, Nedim Lipka, Pranav Maneriker and Srinivasan Parthasarathy. "Open Intent Discovery from Natural Language Interactions." In *Proceedings of the Web Conference (WWW)*, 2020.

Pranav Maneriker, **Nikhita Vedula**, Hussein Al-Olimat, Jiayong Liang, Omar El-Khoury, Ethan Kubatko, Desheng Liu, Krishnaprasad Thirunarayan, Valerie Shalin, Amit Sheth and Srinivasan Parthasarathy. "A Pipeline for Disaster Response and Relief Coordination." In *Proceedings of the ACM SIG Conference on Information Retrieval (SIGIR)*, 2019.

Nikhita Vedula, Pranav Maneriker and Srinivasan Parthasarathy. "BOLT-K: Bootstrapping Ontology Learning via Transfer of Knowledge." In *Proceedings of the Web Conference (WWW)*, 2019.

Hemant Purohit, **Nikhita Vedula**, Krishnaprasad Thirunarayan and Srinivasan Parthasarathy. "Modeling Transportation Uncertainty in Matching Help Seekers and Suppliers during Disasters." In *Proceedings of the ACM SIGIR Workshop on Intelligent Transportation Informatics*, 2018.

Nikhita Vedula, Patrick K. Nicholson, Deepak Ajwani, Sourav Dutta, Alessandra Sala and Srinivasan Parthasarathy. "Enriching Taxonomies with Functional Domain Knowledge." In *Proceedings of the ACM SIG Conference on Information Retrieval (SIGIR)*, 2018.

Nikhita Vedula, Wei Sun, Hyunhwan Lee, Harsh Gupta, Mitsunori Ogihara, Joseph Johnson, Gang Ren and Srinivasan Parthasarathy. "Multimodal Content Analysis for Effective Advertisements on YouTube." In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2017.

Nikhita Vedula and Srinivasan Parthasarathy. "Emotional and Linguistic Cues of Depression from Social Media." In *Proceedings of the ACM Digital Health Conference (DH)*, 2017.

Nikhita Vedula, Srinivasan Parthasarathy and Valerie L. Shalin. "Predicting Trust Relations in Social Networks: A Case Study on Emergency Response." In *Proceedings of the ACM Web Science Conference (WebSci)*, 2017.

Nikhita Vedula, Srinivasan Parthasarathy and Valerie L. Shalin. "Predicting Trust Relations Among Users in a Social Network: The Role of Influence, Cohesion and Valence." In *Proceedings of the ACM SIGKDD Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM)*, 2016.

Fields of Study

Major Field: Computer Science and Engineering

Studies in:

Data Mining	Prof. Srinivasan Parthasarathy
Parallel Computing	Prof. P. Sadayappan
Statistics	Prof. Oksana Chkrebtii

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

The inception and rapid development of online information services and technologies in today's computerized society has inundated the web with unstructured textual information, both formal and informal, across several domains, topics and languages. Some examples include news or research articles, social media interactions, e-commerce platforms, cloud services etc. These vast amounts of human-generated data are treasure troves of valuable knowledge and insights, beneficial to build intelligent, decision-making systems in various domains. Their applications can range from emergency response to computational advertising. However, while accessing or managing these gigantic quantities of digital content might be relatively easy; gaining any concrete, worthwhile or functional insights from this massive online data in a machine-readable format is an onerous undertaking. It requires the learning of relevant knowledge patterns utilizing suitable background context or prior knowledge (*context-aware*), as well as understanding and levering the latent functional intentions of the authors and/or users of such content. Effectively inculcating domain-specific cues into such data-driven systems is also of paramount importance. This is in-turn likely to require inputs from human experts, which will invariably be subjective, time-consuming and expensive. In other words, it is still considerably challenging to:

1. Automatically extract and structurally represent factually accurate information signals from the massive and noisy online digital content, in a context-aware manner with limited manual effort; and

2. Discern the latent behavior, motives, needs, reactions and responses of individuals who create and use multimedia content in various environments.

Jointly studying both of these research directions is the focus of a paradigm called *pragmatics* or *latent pragmatics*. This is the study of the functional intentions and implications behind linguistic content and their variations based on disparate contexts, as defined by linguists Searle, Leech and Jurafsky [142, 175, 281]. In this dissertation, we[1] take steps towards addressing the above two challenges.

The first part of this dissertation tackles the first challenge outlined above, that is associated with latent pragmatic analysis. Namely, we seek to develop knowledge harvesting techniques to extract and organize noisy, unstructured information from numerous domains into easily accessible schema hierarchies with minimal human effort. Such schemas can then act as crucial sources of contextual knowledge for numerous pragmatic applications.

Ontologies, taxonomies or knowledge graphs represent an inherent and effective way of organizing massive amounts of real world information from heterogeneous sources into a structured format. Some examples of large-scale, general purpose taxonomies include Wikipedia Categories[2], Freebase [24] and WordNet [79], useful for various natural language processing applications such as information retrieval [333], document clustering [125], word sense disambiguation [225] and question answering [115]. However, although these

[1] 'We' is stylistically used throughout to represent the findings, opinions and recommendations of the author of this dissertation and all her collaborators.

[2] `https://en.wikipedia.org/wiki/Portal:Contents/Categories`

hierarchies are well developed, they are largely generic with limited lexical and semantic coverage for uncommon languages, less popular entities or highly specialized knowledge domains. It is also prohibitive to continuously augment and maintain them manually with new concepts and relations from newly emerging or rapidly evolving domains such as public health and current affairs. These challenges necessitate the development of automated, scalable techniques to solve the problem of *taxonomy enrichment*. The goal of this problem is to augment a taxonomic hierarchy by accurately placing novel or unfamiliar concepts in it at an appropriate location and level of granularity, avoiding links that are too specific, too general, or contextually related but non-ancestral. For example, based on the Wikipedia article on the *Hurricane Harvey* in August 2017, we would like to link it to coherent, specific categories such as *Category 4 Atlantic Hurricanes* and *August 2017 events in the United States* rather than semantically related but non-ancestral classes such as *Hurricane Irma*, or overly general categories like *Hurricane*. Existing work in the area of automated taxonomy enrichment is primarily highly language-specific [374, 375], domain-specific [254], taxonomy-specific [143, 294, 327] or cannot scale easily to large taxonomies [27, 182, 360].

Apart from general purpose knowledge sources like Wikipedia, WordNet and Freebase, domain-specific ontologies are also valuable resources that formally model the conceptual vocabulary of a given domain. A number of approaches to automate the process of ontology building have been proposed [27,138,182,226,229,328,329,339,348,360,376,387,392,393]. Nevertheless, they only utilize textual corpora and other accompanying information from the specific domain under consideration. They do not leverage the abundant, hierarchically structured knowledge that might be available in functionally and/or *semantically similar* or *related* subjects or domains. This is especially valuable in fields such as epidemiology,

crisis response, bio-medicine and e-commerce, where modeling emerging knowledge in real-time can be frequent and crucial. For instance, there is no semantically coherent ontology associated with the recently surfaced human disease of *Zika fever*, knowledge of which is evolving to-date. It will therefore be highly useful to take advantage of its connections to similar vector-borne diseases like *Dengue* or *Malaria*, for which well organized and annotated information from domain experts is available.

There is another crucial challenge to be tackled while learning hierarchical relationships among multiple concepts from unstructured text. This is the requirement of large amounts of annotated training data in the form of ontological concepts and their corresponding relationships, accompanied by contextual information. It is time-consuming, expensive, and necessitates a significant amount of expert knowledge to categorize associations in niche domains that a layperson is unlikely to know about.

Accurately structuring online information or deriving useful insights from it requires the input information to be factually accurate or reliable. Manual checks and assessments by human reviewers to perform veracity detection are expensive and do not scale to the order of the billions of documents on the web. This has resulted in a plethora of techniques being developed to perform *automated fact checking* [14, 80, 89, 230, 247, 285, 287, 289, 324]. However, most approaches merely focus on *detecting* if a fact is true. They may output numerical veracity scores, which are inadequate and hard to understand for humans. They cannot adequately explain *why* a claim was detected as true or false. Such explanations are desirable because they can (i) provide new and useful insights that can improve the general performance of fact checking; and (ii) help non-experts comprehend the veracity of niche or domain-specific claims. For example, rather than just detecting that the claim *"Due Date is a horror film"* (from [325]) is false, an accompanying explanation such as *"Due Date is a*

2010 American comedy film..." provides a more insightful understanding into the claim's falsity.

The second part of this dissertation seeks to address the second challenge outlined above, associated with latent pragmatics. To this end, we aim to model the intentions and behavioral characteristics of individuals online based on the nature of their created or consumed content, and their social interactions in multiple scenarios. This can in turn provide us a more holistic understanding of human-generated content. We perform this investigation under the umbrella of several paradigms of social good, such as crisis response, mental health and computational advertising.

The problem of recognizing human intentions or *intents* from their text or speech inputs has several concrete applications. Intent detection can highlight, summarize or prioritize user objectives from their conversations, spot action items in emails or meeting transcripts, and reformulate user search queries for better serving their needs. Intent detection is also an essential component of automated dialog response agents or personal assistants (e.g. Amazon Alexa, Apple Siri) that need to parse and interpret human language utterances, in order to effectively and intelligently interact with people and answer their diverse questions.

A comprehensive attempt at dissecting the pragmatics problem must not only investigate the textual cues derived from online information. The language indicators should be supplemented with valuable heterogeneous signals such as valence or emotion, social network interaction details and features from alternate modalities (e.g. video, audio). The advances made in multimedia information retrieval, and the proliferation of user feedback or interaction mechanisms in social media such as comments and ratings present digital videos as an effective tool of studying the problem of multimodal digital content analysis. In particular, commercial advertisement videos in various domains are a good source to

study due to their short length, widespread availability and online presence in the wake of the recent integration of e-commerce infrastructures and web-scale multimedia distribution platforms. A lot of people actively engage with online advertisements by commenting, reacting and responding to them based on their interests, and also getting influenced by them.

We mentioned the importance of the factual accuracy of online information earlier above. Equally important is the reliability or credibility of the people who create and/or disseminate such information, since trustworthy people are likely to generate trustworthy content. To this end, we studied the vital social construct of *trust*, that reflects on the credibility and reliability for the multitude of online participants and data [82]. Affordability, reach, proximity and timeliness make social media such as Facebook and Twitter an attractive resource of observations, activity and information on various topics, domains and events. However, the organizational effectiveness of such resources depends on the filtering, integration and dissemination of *trustworthy* information [262]. Message recipients must trust online users to provide reliable information. They must also rapidly vet and separate noise (inaccurate information from unreliable sources, or ambiguities from reliable sources) from the informative signal. Therefore, a trust ranking of online human sources of information can promote reliance on trustworthy users for various downstream applications.

Peer interactions, including social media interactions play a crucial role in shaping human behavioral outcomes and intentions in modern society. As noted by a recent report from the American Academy of Pediatricians (AAP) [233] and a recent Pew study [249], social media interactions now represent a key communication modality for the vast majority of US adolescents and young adults, and a significant fraction of older adults. Given the pervasive use of social media, a key question then to ask is whether such use departs

significantly from those found in physical (offline) social networks, as studied by sociologists and psychologists for many decades. In other words, does modern online social media communication exhibit similar patterns of user behavior to previously reported studies on offline social engagement?

After a gentle introduction to latent pragmatic analysis and its associated objectives and challenges, we now proceed to present a formal dissertation statement in Section 1.1. This is followed by accomplished work and future research directions in the subsequent sections.

1.1 Dissertation Statement

In this dissertation, we assert that *advances in knowledge extraction and representation as well as pragmatic analysis of author and/or user intentions and behavior, coupled with latent context inference and theoretically grounded domain insights, are critical to practically understand and utilize modern digital content, such as that arising in the web and social media.* To this end, we focus on designing efficient and novel algorithms to understand and represent the vast unstructured digital content, as well as the functional or behavioral motives expressed by creators and/or consumers of said content under various contextual settings, in the presence of no or limited human annotated data. In particular, we seek to answer the following research questions: *How can we automatically extract and represent unstructured and noisy online data in a credible, structured, accessible format (e.g. as ontologies or knowledge graphs) to effectively facilitate search, informative pattern mining, or utilization as stores of background context? How can we automatically recognize generic, dynamically changing functional intentions (intents) and subjects (domains) of human users from their natural language interactions in a domain-agnostic manner? How can we leverage digital content semantics, social interactions, heterogeneous modalities*

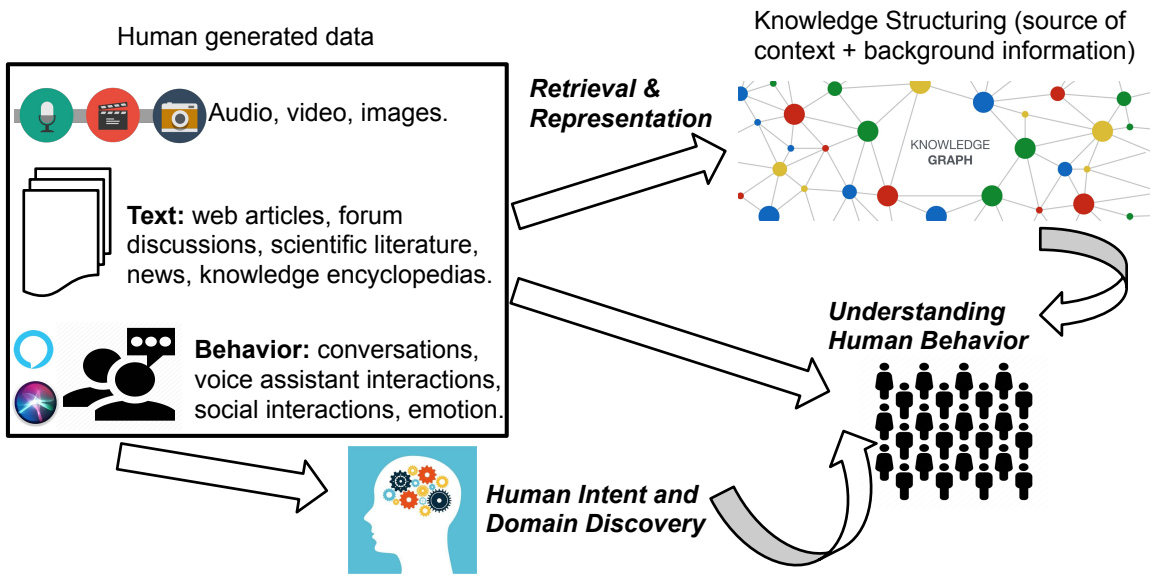


Figure 1.1: An overview of our work on Latent Pragmatic Analysis.

and socio-psychological insights to analyze and explain human reactions, responses, and other aspects of online user behavior?

1.2 Our Contributions

Keeping in mind the above set of questions and our overarching goal of tackling the earlier outlined two broad challenges associated with pragmatic analysis, we now discuss the individual contributions of this dissertation in detail. An overview of our attempt on latent pragmatic analysis is shown in Figure 1.1. We take different forms of human generated digital data as input to our models and algorithms. As output, our models either learn an accurate structured representation of the input content, or understand and model the functional intentions and behavioral characteristics of the content creators and users, under various contextual settings. We highlight our specific contributions in this regard in

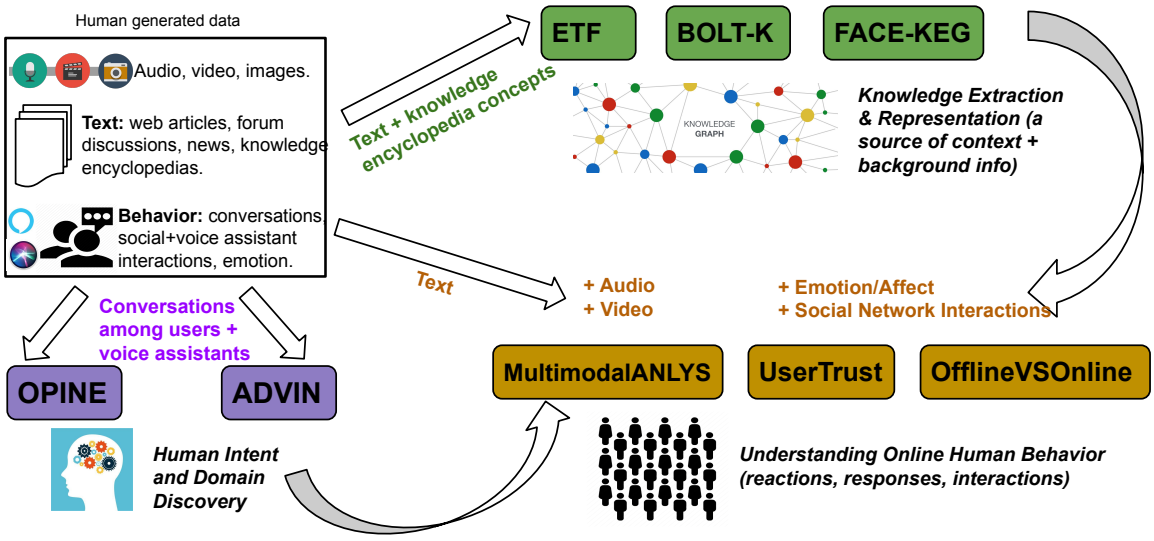


Figure 1.2: Instantiations of our specific efforts towards Latent Pragmatic Analysis: knowledge structuring and representation; information veracity detection; user intent and domain detection; multimodal content analysis; predicting user trust; and comparing offline and online user behavior.

Figure 1.2. There are eight components (in different colored boxes) addressing different types of downstream tasks using varied forms of context, user attributes and digital content.

ETF: Enhancing Taxonomies with Functional Domain Knowledge [339]:

In Chapter 2, we proposed a novel framework, *ETF* (Enhancing Taxonomies with Functional domain knowledge), shown in green color in Figure 1.2, to automatically enrich large-scale, generic taxonomies with newly emerging concepts from resources such as news and research publications, avoiding the traditional method of expert or manual curation. For this purpose, we generated a context for each new and existing concept in the taxonomy, from a corpus of text documents that is associated with them. We trained word-vector embedding models on the existing concepts, and applied these to learn contextual embeddings for the new concepts. We then identified the existing concepts most similar to the new ones, using

a nearest neighbor search in the embedding space. In order to automatically link the new concepts to existing similar taxonomy concepts (i.e. its parents or ancestors), we proposed a ranking model based on a carefully selected set of features. These features leveraged the topological advantages of taxonomies as directed acyclic graphs, lexico-syntactic patterns, semantic properties and external knowledge sources. Extensive evaluations of ETF on large, real-world taxonomies of Wikipedia and WordNet showcased significant improvements compared to state-of-the-art baselines.

BOLT-K: Bootstrapping Ontology Learning via Transfer of Knowledge [337]:

In Chapter 3, we proposed a flexible and generalizable framework *BOLT-K* (Bootstrapping Ontology Learning via Transfer of Knowledge), shown in green color in Figure 1.2, to automatically learn structured ontologies for contemporary novel or emergent sub-domains of rapidly evolving fields such as bio-medicine, epidemiology, e-commerce and crisis response. It uses a long short term memory (LSTM) neural network with attentive pooling. We first obtained all the concepts that would constitute the *target* domain ontology. We subsequently devised an approach based on semantic and topological attributes to identify the concept pairs likely to be connected by an ontological relationship, and eliminate the remaining spurious combinations. To address the issue of limited expert-labeled training data for the newly emergent target domain, we utilized publicly available textual corpora and ontological information from a functionally similar *source* domain. This bootstrapped the task of learning ontologies for the novel target domain by adapting and *transferring* existing knowledge from the related source domain ontology. We also employed data augmentation techniques to generate additional training examples for the target domain. We trained BOLT-K jointly on the target and source hierarchy information, by sharing the hidden feature representations and appropriate model parameters among them. Finally, we

predicted ontological relationships between the concepts of the target domain, to construct an ontology for it.

We extensively evaluated BOLT-K on several real-world datasets, highlighting the transferability of concepts across comparable subjects. We also showed BOLT-K's capability in detecting novel types of relationships that were unseen during training.

FACE-KEG: Fact Checking Explained using Knowledge Graphs [342]:

In Chapter 4, we proposed a framework *FACE-KEG* (FAct Checking Explained using KnowledgE of Graphs), shown in green color in Figure 1.2, to solve the problem of *automated fact checking*. That is, our technique detected if a given piece of information is factually correct or incorrect, by jointly modeling structured and unstructured contextual knowledge. Our framework further learned explanations justifying the veracity of the input claims from the interrelated perspectives of the claim content, suitable background context and structured conceptual knowledge relevant to the claim. To the best of our knowledge, this was the *first* attempt in the literature to explain fact checking by directly generating human-readable textual explanations clarifying the veracity of input facts. FACE-KEG first built a knowledge graph and retrieved unstructured background context pertinent to each input claim. A novel graph transformer network and a bidirectional RNN network were employed to encode the knowledge graph and textual context respectively. This was followed by jointly training a classifier that predicted if the claim is true or not, and a decoder that learned to generate an abstractive natural language explanation clarifying the veracity of the claim. We extensively evaluated our approach using a mixture of both automated and human evaluation measures, and achieved significant gains over state-of-the-art baselines.

OPINE: Open Intent Extractions from Natural Language Interactions [185, 336]:

In Chapter 5, we defined a novel problem of *open intent discovery*, and proposed a neural network based framework called *OPINE* (OPen INtent Extraction) to solve it (shown in violet color in Figure 1.2). The goal was to automatically discover actionable user intents explicitly mentioned in natural language, *without* prior knowledge of a comprehensive list of intent classes that the text utterance may comprise of. In other words, OPINE could recognize instances of intent types that it had never seen before. Unlike prior literature, our proposed approach modeled the open intent discovery problem as a *sequence tagging* task. We developed a neural model consisting of a Conditional Random Field (CRF) on top of a bidirectional LSTM with a multi-head self-attention mechanism. To ensure its effectiveness across several task domains or fields even in the absence of sufficient labeled training data, OPINE represented all kinds of user intents extracted from the textual input in a consistent and generalizable format, independent of their domain. We further employed adversarial training at the lower layers of our model, and unsupervised pre-training in the target domain under consideration. Extensive experiments on multiple real-world datasets showed the accuracy and efficacy of our approach. Moreover, commonly used intent-labeled datasets in dialog research such as SNIPS [61] or ATIS [62, 122] largely have concise, coherent and single-sentence texts. They are not very representative of complex, real-world dialog scenarios which could be verbose and ungrammatical, with intents scattered throughout their content. Therefore, we developed a large dataset with 25K real-world utterances from the online question-answer forum of Stack Exchange. They spanned several genres and were curated for intents by crowd workers.

ADVIN: Automatically Discovering Novel Domains and Intents from User Utterances [335]:

Our previous work on OPINE could only identify user utterances containing *actionable* intents, and also could not identify the domains of the input text utterances. In Chapter 6,

we sought to tackle these two limitations, by solving the problem of novel user intent and domain discovery (shown in violet color in Figure 1.2). We attempted to bridge the gap between the two challenging yet realistic tasks of (i) identifying utterances belonging to novel, generic intents and/or domains, not seen before during model training, and (ii) organizing the newly discovered intents and domains into a taxonomy. We proposed a novel, three-step framework called *ADVIN* (Automated Discovery of noVel domaIns and iNtents). It automatically discovered user intents and domains in massive, unlabeled text corpora, *without* any prior knowledge about the intents or domains that the text may comprise of. Our method first leveraged the pre-trained multi-layer transformer network, BERT [69], to determine if an utterance is likely to contain a novel intent or not. ADVIN next used unsupervised knowledge transfer to discover the latent intent categories in the earlier identified utterances. Finally, ADVIN hierarchically linked semantically related groups of newly discovered intents to form new domains. We extensively evaluated ADVIN on four public benchmark datasets and real-world data from a commercial voice agent, and significantly outperformed baselines across various empirical configurations.

Multimodal Analysis of Digital Media Content in the Advertising Domain [345, 347]:

In Chapter 7, we proposed and implemented a computational framework that analyzes digital video content in a multimodal fashion, utilizing three modalities of video, audio and natural language. In particular, we performed a predictive analysis of content-based features extracted from online commercial advertisement videos (shown in brown color in Figure 1.2). Our proposed framework extracted multi-dimensional temporal patterns from the content of advertisement videos using multimedia signal processing and natural language processing tools. The pattern analysis part employed an architecture of cross modality feature learning. Data streams from different feature dimensions were employed to train separate neural

network models. These models were then fused together to learn a shared representation. Subsequently, a recurrent neural network model trained on this joint representation was utilized as a classifier for predicting effectiveness of the input advertisement among its target audience group. We validated our approach using subjective ratings from a dedicated user study, the text sentiment strength of online viewer comments, and a viewer opinion metric of the likes/views ratio of each advertisement from the video-sharing website *YouTube*. We investigated the interplay of complex factors contributing to advertisement success among target audience groups such as the mix of reason and emotion, synergistic interactions between music and narrative speech, spatio-temporal organization of video shots, and brand label mentions. Based on the predictive patterns identified by our model, we elicited a useful set of auditory, visual and linguistic patterns to aid advertisement effectiveness. These are not only strongly correlated with the proposed evaluation metrics, but can be readily implemented in the design and production processes of commercial advertisements.

Predicting Trust Relationships in Online Users [343, 344]:

Seeking explicit assessments of trust among users at scale is impractical in an online social network setting. Instead, grounded in appropriate social and psychological theories, in Chapter 8, we developed an unsupervised model to learn a representation for the human socio-behavioral trait of trust in an online social network setting (shown in brown color in Figure 1.2). We integrated the implicit factors of social influence exerted by each user over their social network, users' inherent structural roles obtained from their underlying network topology, and the semantic intentions and affective valence extracted from their short, informal, asynchronous texts. A key finding was the importance of modeling influence and affective valence in such exchanges and their role in detecting stable trust relationships.

We extensively evaluated these ideas and demonstrated significant gains over competitive baselines across multiple social media datasets drawn from various scenarios.

Emotional and Linguistic Cues of Online Behavior Patterns Focused on Clinical Depression [341]:

In Chapter 9, we investigated if online communication among users and their social network neighborhood exhibited behavioral patterns similar to those from offline social engagement, with a focus on *clinical depression*. This is shown in brown color in Figure 1.2. To facilitate the analysis of our observational study, we examined network effects related to user participation, user engagement and interactions with their ego-neighborhood. We defined network participation features to include both passive (content users are exposed to or receive from other users) and active participation (content created by users themselves). We defined network experience features to include both content (e.g., linguistic cues, emotion) and relational dynamics (e.g., conflict/support, influence) of network embedded interactions. We also examined neighborhood effects and analyzed key statistics of the neighborhood such as size, centrality and affinity to form clusters or communities. Our study included both depressed users and their ego-net(s) as well as non-depressed users (control group) and their ego-net(s). We observed significant deviations in the behavior of depressed users from the control group. Based on our observations, we then described an approach to extract relevant features and build a high accuracy classifier to predict the onset of clinical depression in social media users.

1.3 Conclusions and Future Work

The long-term goal of this dissertation is to advance knowledge acquisition, representation and inference systems as well as facilitate a multi-faceted, pragmatic understanding of

the users and/or authors of such knowledge. We believe that analyzing and transforming massive sources of heterogeneous data in diverse contexts, with theoretically grounded interdisciplinary domain insights in a mutually reinforcing manner, will greatly help in future towards solving the broader goal of latent pragmatic analysis. Another important objective that we attempt to address throughout this dissertation is developing state-of-the-art algorithms that effectively utilize no or minimal human annotations as part of model development and training. All these aspects are sure to play a crucial role in real-world pragmatics applications for diverse fields such as marketing, crisis response, conversational search and public health. We conclude this dissertation and present some interesting and important directions of future research in Chapter 10.

The rest of the dissertation is structured as follows. Chapter 2 outlines our method, ETF, of automatically enhancing the coverage of general-purpose taxonomies with domain specific information. Our technique, BOLT-K, that learns ontologies for newly emerging target domains with limited annotated data by levering existing knowledge from related domains is presented in Chapter 3. Chapter 4 details our technique, FACE-KEG, to predict the veracity or falsity of input textual claims, and generate human-readable explanations for its predictions. In Chapters 5 and 6, we present two frameworks, OPINE and ADVIN respectively, that automatically discover novel actionable intents (OPINE) as well as both novel intents and domains (ADVIN) from user utterances. We describe our multimodal technique of analyzing digital media content in the advertising domain in Chapter 7. An unsupervised technique of predicting trust relationships among social media users is discussed in Chapter 8. Chapter 9 describes our empirical analysis of online user behavior patterns with a focus on clinical depression. Finally, Chapter 10 concludes this dissertation and discusses potential future research directions.

Chapter 2: ETF: Enriching Taxonomies with Functional Domain Knowledge

Beginning with this Chapter, our first three Chapters 2, 3 and 4 focus on addressing the first challenge associated with latent pragmatic analysis as identified in Section 1. This refers to extracting and organizing unstructured information into easily accessible sources of contextual knowledge. Enriching knowledge resources with new information to better model the "changing world" presents two-fold challenges: (1) Detection of previously unknown entities or concepts, and (2) Insertion of the new concepts into the knowledge structure, respecting the semantic integrity of the created relationships. To this end, we propose a novel framework, *ETF*, to enrich large-scale, generic taxonomies with new concepts from resources such as news and research publications. ETF learns a high-dimensional embedding for the existing concepts of the taxonomy, as well as for the new concepts. During the insertion of a 'new' concept into an existing taxonomy, this embedding is used to accurately verify that a concept is new to the taxonomy, and identify semantically similar neighborhoods within the existing taxonomy structure. The potential *parent-child* relationships linking the new concepts to the existing ones are then predicted using a set of semantic and graph features. We also demonstrate that ETF can accurately categorize newly emerging concepts and question-answer pairs across different domains.

2.1 Introduction

Human knowledge is inherently organized in the form of semantic, content-specific hierarchies or taxonomies such as Wikipedia Categories[3], Freebase [24] and WordNet [79]. However, although these hierarchies are well developed, they are largely generic and laborious to augment and maintain, with new concepts and relations from newly emerging, or rapidly evolving domains such as public health and current affairs. These challenges necessitate the development of automated, scalable techniques to solve the problem of *taxonomy enrichment*, i.e., to augment a taxonomic hierarchy by accurately placing novel or unfamiliar concepts in it at an appropriate location and level of granularity. For example, based on the Wikipedia article on the *Hurricane Harvey* in August 2017, we would like to link it to coherent, specific, parent categories such as *Category 4 Atlantic Hurricanes* and *August 2017 events in the United States* rather than semantically related but non-ancestral classes such as *Hurricane Irma*, or overly general parent categories like *Hurricane*.

Existing work in the area of automated taxonomy enrichment is either highly language-specific [374, 375], domain-specific [254] or cannot scale to large taxonomies [27, 182, 360]. Many techniques depend on the unique synset structure specific to WordNet [143, 294, 327], cannot generalize to other taxonomies, and can only identify a single subsuming category for new concepts. Even with these limitations, not all attempts at taxonomy enhancement have succeeded [254]. In contrast, we identify that the main challenge for this task is to find a computational measure that can proxy the logic of semantic subsumption, independent of the language and knowledge domain, in order to automatically find good parents for new concepts. To address this fundamental challenge, we propose a combination of a carefully selected set of highly effective graph-theoretic features and semantic similarity based features

[3] `https://en.wikipedia.org/wiki/Portal:Contents/Categories`

leveraging external knowledge sources. We show that this combination measure can predict the links to the parents of new concepts with high accuracy. Interestingly, our measure does not require any assumptions on the number of parents for each new concept. Moreover, our evaluation methods are fully automated using publicly available data and don't rely on domain experts or manual judgment, unlike prior work [33, 179, 226, 294, 357].

Intuitively, we believe that ETF is similar to how a human would approach the task of adding a new concept to a taxonomy. To complete such a task, one might: (i) find a set of existing concepts that are related to the new one, (ii) rank each concept in the union of ancestors of this set, i.e., the closure via hypernymy/parent relations, and; (iii) link the new concept to a few selected top-ranked parents.

To formalize the intuitive method we just outlined, ETF first constructs a vector representation for new concepts. Here, we use a representation that aggregates two kinds of embeddings formed from the context of the concept. We observed that the embeddings complement each other and so the aggregated vector provides a good measure for semantic similarity, that can identify the nearest neighbors of the new concept in the taxonomy. With this representation, the search for potential parents is restricted to the (small) set of ancestors of these nearest neighbors. Experiments show that on test concepts from the Wikipedia category taxonomy with more than 5 million concepts, on average this set contains a few thousand nodes and covers 83% of the true parents of the test concepts. Finally, we develop algorithms to rank the ancestors in this set, selecting only those above a global scoring threshold to be the parents of the new concept. Thus, the key contributions of our work are:

- We develop a novel, fully automated framework, ETF, that generates semantic text-vector embeddings for each new concept. These embeddings allow us to find semantically related concepts in the existing taxonomy, which in turn allows us to extract the ancestors of these related concepts.

- We propose the use of a learning algorithm that combines a carefully selected set of graph-theoretic and semantic similarity based features to rank candidate parent relations. This ranker accurately links new concepts to good candidate parents by ranking the ancestors of their semantic neighbors.

- We show via two case studies that ETF can accurately categorize new concepts from evolving real-world domains, as well as new questions and answers from Quora[4].

2.2 Related Work

The problem of automatic taxonomy induction, i.e., effectively reconstructing an entire taxonomy, has received a lot attention in the literature. Supervised and unsupervised machine learning techniques based on co-occurrence analysis [357], clustering [179], graph construction and traversal [226] and distributional similarity [374, 375] have been used to solve this problem. Linguistic pattern-matching based approaches [121, 226, 294, 379] have been employed to discover relations between a term and its hypernyms. Using word embedding based techniques for identifying relations to recreate taxonomies has also gained popularity in recent years [87, 315, 329, 389]. Some of these techniques [87, 315] suffer from low accuracy in taxonomic relation prediction, while others [389] do not generalize to unseen relation instances. The method proposed by Tuan et al. [329] appears to tackle these issues, however the input to its training phase requires hypernym-hyponym pairs to occur

[4] `http://quora.com`

in a sentence, which is quite unlikely in the case of Wikipedia-style concept and category names. We also utilize term embeddings as part of our approach to enhance taxonomies (Section 2.4), however we combine it with well-designed graph-based and semantic features to maximize performance.

There already exist fairly accurate, general-purpose knowledge bases that have been painstakingly curated by experts or via crowdsourcing. Hence, rather than constructing new taxonomies from scratch, our work leverages these existing taxonomies and enhances them with new information. This problem of automatically enhancing the coverage of extant taxonomies has primarily focused on enhancing the WordNet taxonomy. Toral et al. [327] extended WordNet with about $320,000$ named entities and their relations, derived from Wikipedia categories and articles. Widdows [365] developed a method to place an unknown word where its neighbors are most concentrated, an idea also leveraged by our work. But his work used part-of-speech tagging to find the nearest neighbors, which is unlikely to work with the Wikipedia-style concept names that ETF can handle. WordNet augmentations have also been proposed for the domains of technical reports [353], medicine [33, 78], and architecture [19]. However, many of these efforts suffer from low accuracy, require part-of-speech tagging, and depend on the category structure peculiar to WordNet. Thus, they cannot easily generalize to other taxonomies, unlike our approach. These works also need human judges for evaluation, while ETF does not. Task 14 of SemEval 2016 [144] is based on extending WordNet with concepts from various domains such as religion, law and finance. We evaluate the performance of ETF on this task in Section 2.5.

Efforts similar in motivation to our work have attempted to extend a generic taxonomy with domain specific knowledge. Ancestor-descendant relationships and hypernym patterns have been extracted from large text corpora such as Wikipedia [121, 306], to enhance

taxonomies. Yamada et al. [374, 375] augmented the Japanese Wikipedia and WordNet with new Japanese terms. They first found similar words from the Wikipedia database, scored the hypernyms of these words, and selected the top-scored hypernym as the output. However, their scoring technique is heavily dependent on verb-noun dependencies found in the Japanese language, that are not usually seen in English. These works also attach the new word to a single parent, and require human judges for evaluation.

Recently, the related problem of *knowledge graph completion*, i.e., predicting relations between entities using supervision from an existing knowledge graph, has received attention in the literature [27, 182, 360]. These techniques predict missing links in knowledge graphs by learning entity and relationship embeddings, based on the idea that the relation between two entities corresponds to a translation between their embeddings. We compare ETF against a state-of-the-art link prediction method, *TransR* [182], in Section 2.5. However, we note that the processing time of these methods is of the order of days for large taxonomies (e.g., the Wikipedia category taxonomy).

2.3 Problem Formulation

We now formally define our taxonomy enrichment problem. A *taxonomy* $T = (V, E)$ is a directed, acyclic graph (DAG) where the set of vertices or nodes V consists of all its hierarchically organized entities and categories, and the set of edges E represents the node relationships. The direction of an edge in the graph hierarchy is from a specialized node to a more generic node that subsumes it. *Entities* in T are designated by nodes that do not have any incoming edges to them, i.e., they have an in-degree of 0. Since these are at the lowest or most specialized 'level' in their respective sub-hierarchies in T, we also refer to them as *leaf* nodes. We use the term *concept* to refer to any kind of node, leaf or otherwise,

in T. The *ancestors* of a node v in T are the set of nodes $A(v)$ in T reachable from v. v is thus considered a *descendant* of all nodes belonging to $A(v)$. The *k-hop neighborhood* of node v is the set of all nodes that are reachable from v via a path of length at most k in T. The immediate ancestors of v, reachable in a single hop, are called its *parents* or *hypernyms*, whereas v is called a *child* or *hyponym* of its parents.

Though the taxonomy graph *should* be a DAG, it may not be initially free of cycles, due to human error, or from unifying collaborative efforts from disparate sources while constructing the taxonomy. Since they can lead to hierarchical inconsistencies, we remove cycles from our taxonomy via known methods [308, 310]. We assume that existing concepts have some text associated with them. For example, in the Wikipedia category taxonomy, the Wikipedia pages provide the text. However, if the taxonomy was created from a document corpus, an aggregation of the context around each mention of the concept can be used as the associated text for that concept. The inputs to our problem therefore are:

(1) A corpus of unseen, unlabeled, unstructured text documents containing a set of new concepts X. Ideally, each document would provide a definition for exactly one new concept. This definition can be as short as a couple of sentences. However, it is more likely that each document may refer to one or more of the new concepts, without specifically defining these new concepts. Our approach can handle either scenario, though we focus primarily on the more difficult latter case. Here we expect that the corpus contains many (i.e., at least 10) references to each new concept.

(2) A taxonomic hierarchy of categories and entities, T.

(3) A corpus of text documents D associated with the concepts present in T. Note that a document can be as small as a single sentence of text. These documents need not be formal definitions of the concepts in T, and may even be automatically created.

Each concept $x \in X$ must be checked to determine if it is already present in T. If not, a solution inserts x into T by outputting a set of semantically appropriate parents in T for x.

Discussion: We emphasize that our primary focus in this work is finding the candidate parents for x, rather than the first step of verifying whether x is already present in X. We also note that our problem formulation implicitly assumes that each new concept will be inserted into the taxonomy T as a *leaf* node. This follows since we find a set of parents for a new concept, but do not attempt to find its children that may exist in the taxonomy. The decision to restrict our search to parents was taken due to practical reasons:

(1) Taxonomies tend to have many more leaves than 'non-leaf' concepts. Hence, finding candidate children is often prohibitively expensive. A new concept may have a few thousand candidate parents, but potentially millions of candidate children.

(2) Large knowledge bases (e.g., WordNet and Wikipedia) have well developed categories. Thus, *most* new concepts being added are likely leaves or will eventually be extended top-down to become roots of sub-graphs, so leaf insertions are, by frequency, *the most important* case to consider.

(3) We acknowledge that a bottom-up procedure, in which the taxonomy is extended by inserting new categories above a set of leaves is also possible. Using ETF, it *is* possible to insert new categories into the taxonomy, by linking them to appropriate parents as well as children. However, beyond the cost of finding and examining all candidate children, there are other drawbacks to this approach: (i) category insertions add a temporal dimension, as the order in which entities and categories are inserted matters. This temporal aspect not only complicates the experimental set-up, but also the evaluation and presentation of the results; (ii) category insertions can introduce cycles into the taxonomy, which must then be detected and resolved.

We thus focus only on leaf (i.e., entity) insertions in this work.

2.4 The ETF Framework

In this section, we describe our framework, ETF (Figure 2.1). We first learn a representation for the new and existing concepts in the taxonomy (Section 2.4.2). For each new concept, we leverage its representation to identify the entities in the taxonomy to which it is most similar. We then narrow down the search for the new concept's potential parents to the set of ancestors of the similar entities. Using a combination of graph-theoretic and semantic features as a proxy for semantic subsumption (Section 2.4.3), our framework filters and ranks these ancestors, and adds appropriate links to the taxonomy.

2.4.1 Finding Concepts and Taxonomic Relations

This is the first step of our algorithmic pipeline. We acquire the entities and categories from the given taxonomic structure and utilize their ancestor-descendant relationships to construct a DAG T as mentioned in Section 5.2. Generally, all entities (leaf nodes) and many categories (non-leaf nodes) in T are associated with text documents, which make up the corpus D. The next task is to obtain the novel concepts to be integrated into T. For instance, these could be the name of an emergent disease, a new organism species, or a recently passed law. In many cases, what is available to us is only a text corpus such as news reports, laboratory records or research articles containing descriptions of unfamiliar concepts or ideas. Off-the-shelf Named Entity Recognition algorithms [131, 199, 223] can be applied to locate and extract named entities from the text. This step must be followed by segregating the novel entities from those already present in T, which, as mentioned, is a challenging problem in its own right and not the focus of this work. However, we do perform a preliminary investigation of the capability of our framework to verify that

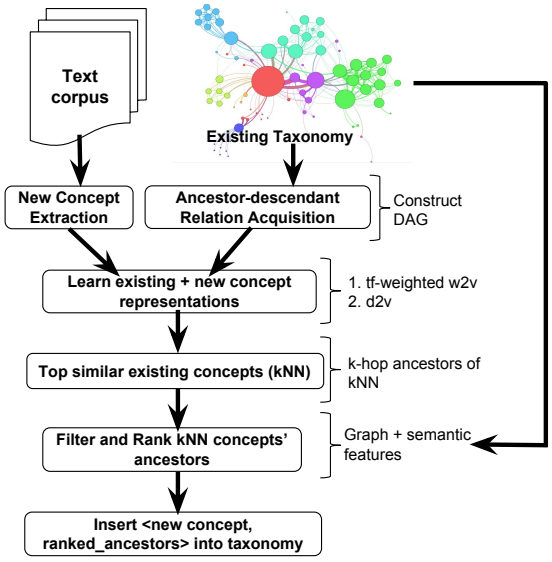


Figure 2.1: An overview of the pipeline of our ETF framework.

these new concepts indeed do not already occur in the current taxonomy (Section 2.5.1.1). For later steps though, we assume that the new concepts to be incorporated into the extant hierarchy are precisely known beforehand.

2.4.2 Learning Concept Representations

After acquiring the relevant concepts and their relationships, we propose to learn a meaningful, text-based embedding for the existing and new concepts. Recent literature (e.g. [87, 341, 345, 389]) has seen an extensive use of learning and employing high-dimensional word embeddings for diverse applications. We build upon the highly effective Skip-gram variant of word2vec embeddings [213]. It minimizes the log loss of predicting the context of an instance using its embedding as input features. Formally, let (i,c) be the set of pairs of entity i and context c. The loss function is then:

$$-\sum_{(i,c)} \log p(c|i) = -\sum_{(i,c)} \left(\mathbf{w}_c^T \mathbf{e}_i - \log \sum_{c' \in C} \exp(\mathbf{w}_{c'}^T \mathbf{e}_i) \right)$$

where $\mathbf{w}$'s are the parameters of the Skip-gram model, $\mathbf{e}_i$ is the embedding of entity i and C is the set of all possible contexts. Under Skip-gram, for each training pair (i, c), the term i is the current term (entity) whose embedding is being estimated, and the context c is made up of the words surrounding i within a fixed window size.

We first replace all occurrences of entity names (i.e., the specific, potentially non-unique, noun phrases that are associated with that entity) in the corpus D of existing text documents. This can be done by disambiguating the noun phrases [202] to the correct entity, and representing the entity by a canonical unique term. Note that, in the case of Wikipedia, these links have already been established by human annotators, so the disambiguation step is not necessary. We then train a doc2vec (distributed memory or DM version of Paragraph Vector) [173] model with negative sampling [213] on D, in the joint space of terms and entities (each represented by a canonical term), that gives us vector representations for the documents in D. Note that this is different from the past works on embeddings that exclusively work in the space of terms, as the step of replacing all the (possibly non-unique) set of noun phrases that refer to a specific entity by a canonical identifier removes a great deal of ambiguity from the corpus. Doc2vec has a similar loss function as specified above except for an additional 'paragraph token' that is pre-pended to the beginning of the context. The DM mode of doc2vec simultaneously learns word2vec word vectors and document vectors in the same space during training, thereby enabling easy aggregation of these vectors.

To get the representation of an entity, we add a *tf*-weighted sum of the word2vec embeddings of its context terms to the doc2vec representation of its associated document. The intuition behind this representation is that word embeddings effectively capture the semantic relationships between individual words in a document, while document embeddings summarize the overall semantics of their constituent words. By aggregating the

two vectors, we hope to maintain the representational effectiveness and non-sparsity of doc2vec, while also preserving individual word semantics. Furthermore, we find that the document representation constructed from word2vec is biased towards highly frequent terms or those that express functions. However, the doc2vec representation is more affected by low frequency, content-rich words, since the more frequent words are likely to be chosen as negative samples. We empirically demonstrate the benefits of aggregating these two kinds of embeddings in Section 2.5 (Figure 2.2).

After creating embeddings for the existing concepts in T, we next learn representations for the new concepts to be inserted into T. This is essentially the task of inductive learning [364], which generalizes an existing learning model to produce representations for unseen data items. However, it is non-trivial to achieve this in an unsupervised setting. Prior work has done this by either incrementally re-training the learning model [248] which is expensive, or using semi-supervised techniques based on neighborhood attributes [364, 380] to learn embeddings of unobserved instances. To accomplish this in an unsupervised manner, we first generate a context c for each new concept, composed of the frequent terms based on their *tf*-score, aggregated from all documents in the corpus where the new concept occurs. We infer a doc2vec embedding using c via gradient descent, by keeping the current doc2vec model constant [173]. We then approximate the embedding of the new concept by adding the *tf*-weighted sum of the word2vec embeddings of its context terms to its inferred doc2vec embedding.

2.4.3 Filtering and Ranking Potential Parents

With the embeddings of the new and existing concepts in hand, we now must determine the best parents for each new concept within T. To this end, for each new concept, we find

the k-nearest neighbours (kNNs) that are most similar to it in the vector embedding space. We hypothesize that the best parents for the new concepts are highly likely to be in common with the ancestors of these kNNs. What remains now is to rank the candidate parents for the new concept. For this pupose, we build a learning-to-rank model that uses topological features from the taxonomy's graph structure, and semantic features derived from the text.

Specifically, we used LambdaMART [34] with a set of training examples consisting of relevant and non-relevant parents of the existing concepts in T. Note that we tried seven other learning-to-rank models (e.g., SVMRank [139]), but found that LambdaMART gave the best performance. The test set consists of potential parents of a set of new concepts, disjoint from the training examples, to be ranked. As mentioned earlier, these candidate parents are the ancestors of the kNNs of the new concepts. 'Relevant' or positive training examples are therefore all those categories that are parents of existing concepts belonging to the training set. 'Non-relevant' or negative training examples are categories that have been randomly sampled from the ancestors of the kNNs of the new concept. We take an equal number of positive/negative training examples for each concept.

We studied fifteen topological or graph-based and semantic features, before selecting a subset of six features that provides a good coverage of various structural and content-based properties, coupled with better performance of the learning-to-rank model. The actual values of the features can either be a score defined for a pairwise property (usually the pair of a new concept and its potential parent), or for an individual property (of the potential parent). The topological features we studied include measures relating to graph centrality, and path traversals over the taxonomy DAG. We also studied semantic features based on the contextual difference between new concepts and their prospective parents, their co-occurrence, contextual term overlap, term overlap with respect to certain parts-of-speech,

and lexico-syntactic patterns such as hypernym patterns or sibling patterns [121,294]. Below, we describe the features most beneficial for our ranking model.

2.4.3.1 Graph-based Features

We found three graph-based features to be particularly effective in ranking candidate parents: Katz similarity, random walk betweenness centrality and an information propagation measure. The first two are based on an undirected graph obtained by taking the nearest neighbor nodes and their ancestors up to a certain depth. We then take all the directed edges between these nodes from the taxonomy and insert them into the undirected graph.

Katz Similarity: Intuitively, there should be many short paths between the new concept and its correct parents. Katz similarity [148] captures this intuition and hence turns out to be a highly discriminative indicator for semantic subsumption. We compute the Katz similarity between each nearest neighbor x and potential parent p, and average them for each potential parent as: $KS(x,p) = \sum_{l=1}^{l_{max}} \eta^l \cdot |paths_l(x,p)|$, where $|paths_l(x,p)|$ is the number of paths of length $l \leq l_{max}$ between x and p, and η is an attenuation parameter ($0 < \eta < 1$) that ensures that shorter paths are weighted higher.

Random Walk Betweenness Centrality: A good parent should generalize the most similar neighbors of the new concept well. One way to capture this is by considering random walks from the neighbors and measuring the betweenness centrality of the candidate parents based on these random walks. A general random walk betweenness centrality measure [228] roughly measures how often a node is traversed by a random walker going from any node in the network to another. Hulpus et al. [132] proposed a focused random walk betweenness centrality measure that focuses on the paths between all pairs of a pre-defined subset of

nodes. Our feature is akin to this focused measure, where the pre-defined subset is the set of neighbors of the new concept.

Information Propagation Score: One problem with considering undirected graphs for concept generalization is that it can result in topic drift and hence, the resultant measure becomes noisy. To address this we consider another feature, namely an *information propagation score*, that propagates the weight from the neighbor nodes upward along directed edges to more general concepts, following the directional traversal on the hypernym edges.

Consider the set of nearest neighbor entities for the new concept, where each entity has a weight associated with it, i.e. the similarity between itself and the new concept to be inserted into T. These weights are then propagated upwards towards the ancestor nodes in T. This propagation ensures that highly central parent nodes lying on many neighbor-to-root paths accrue large weights. To avoid over-generalization, each intermediate node decays the weights by a multiplicative factor $(1-\delta)$, which we call the *decay factor*.

However on noisy, real-world taxonomies such as Wikipedia categories, additional issues may arise. First, there can be a great disparity in the number of parent categories $P(v)$ of each node v. This causes the parents in $P(v)$ to obtain low weight values if $P(v)$ is large, or large values if $P(v)$ is small, if the propagated weights are uniformly split among all parents. We thus introduce a parameter $\alpha < 1$ that enables us to propagate a proportion $\frac{1}{p(v)^{\alpha}}$ of the weight at each step. Second, while most edges in T are likely to be between nodes at similar levels of granularity, some may be direct connections between highly specialized and highly generic concept nodes. This can cause a generic node to get a higher weight than its counterparts at similar levels. We thus additionally penalize highly generalized nodes far away from leaf nodes by a factor β.

Let $p(v)$ be the number of parents of node v in $T(V,E)$, and N_x be the set of nearest neighbor entities of a new concept x. The initial weight of each entity v in N_x is given by $w_0(v)$, which is the cosine similarity between v and x in the embedding space, and 0 otherwise. Let $\text{ld}(v)$ be the length of the longest path from a leaf node in T to v. The information propagation score $IP(v)$ is defined as the total weight passing through v from the leaf level, given by:

$$IP(v) = \begin{cases} w_0(v), & v \in N_x. \\ \sum\limits_{(u,v)\in E} \frac{(1-\delta)IP(u)}{p(u)^{\alpha} e^{\text{ld}(v)\beta}}, & v \notin N_x; u, v \in V. \end{cases}$$

2.4.3.2 Semantic Features

In addition to the graph features, we found the following semantic features to be highly discriminative:

Ancestor-Neighbor Similarity: For each novel concept, we compute the pairwise term overlap between the text document linked to its potential parent under consideration, and that associated with each of its nearest neighbor entities. We then take the average of these overlap values. The importance of this feature stems from the fact that a good ancestor should generalize the properties of as many entities highly similar to the target concept as possible. For this feature, we only consider those ancestors or parents in the taxonomy DAG T that have text associated with them.

New concept-Ancestor Similarity: Since a satisfactory parent of a new concept is a good generalization of it, it is quite likely to have similar text as the new concept itself, i.e. high textual overlap with the new concept. This feature thus computes the Jaccard similarity between the occurrences of textual terms in the document associated with the new concept, and those associated with the category or ancestor document. As earlier, we only consider those ancestors in T that have text associated with them.

Pointwise Mutual Information (PMI): A parent of a concept has a high probability of co-occurring or being mentioned together with it. Yang and Callan [379] have shown this feature to be highly successful in indicating semantic relations between terms. PMI measures the co-occurrence of a new concept x and its potential parent p via the pointwise mutual information between them: $PMI(x,p) = \log\left(\frac{num(x,p)}{num(x)\cdot num(p)}\right)$, where $num(t)$ (or $num(t_1,t_2)$) is defined as the number of occurrences of a particular term t (or co-occurrences of a pair of terms), either in a set of sentences, or documents which can be from a large corpus such as Wikipedia or the web.

We tested our approach by computing the PMI using two such corpora of documents. First, we investigated the PMI between each new concept and each of its prospective parents by checking their co-occurrence in Wikipedia. However, many peculiar concept and category names do not occur in text documents, hence we could not gain a performance boost with this feature. The second kind of PMI that we compute is by checking the co-occurrence of the new concepts and their potential parents in all pages on the web. We thus compute the value of $num(t)$ by querying the Bing Search API[5] to find the number of search results or pages on the web containing the respective term t. Evidently, we only consider those potential ancestors in T that have at least one web page in which they occur. Henceforth, all references to the feature 'PMI' refer to the PMI value computed using the Bing Search API.

Once the ancestors have been ranked in order of their relevance to the new concept, we connect the new concepts to their 'r' top-ranked ancestors and evaluate the resulting taxonomy (Section 2.5).

2.4.4 Parameter Tuning

In this section, we describe how various parameters were tuned.

[5] `https://azure.microsoft.com/en-us/services/cognitive-services/`

Embedding Parameters: We trained 200 dimensional embeddings using the gensim implementation [259] of doc2vec. We used its distributed memory version since it was found to outperform other variants [173], and a context window size of 10.

Number of Neighbors and Depth of Ancestors: These values are selected according to an accuracy/computational cost trade-off. We choose the top 50 nearest neighbor entities to the new concept, whose ancestors we want to evaluate, since we found that additional noise is added to the set of potential ancestors as this value increases. Further, rather than considering *all* the ancestors, we only take into account those ancestors that are reachable from the nearest neighbor entities in at most 3 hops in T. We found that ancestors up to three levels above the neighbors of the new concept span a large portion of T without being too generic or too specific.

Parameters of Graph Features: For the parameters α, β and δ of the information propagation feature, we perform an exhaustive grid search on a sample of our training dataset, and find the best empirical performance with $\alpha = 0.75$, $\beta = 0.005$ and $\delta = 0.005$. We assign the value of the probability parameter q used for computing the random walk betweenness centrality equal to the value of α, i.e. 0.75. For the Katz similarity feature, we use $\eta = 0.2$.

Predicting the Number of Parents: We learn a global ranking threshold based on the rank scores received by the correct, ground truth parents of the training set. The new concept is thus connected to all those parents receiving a rank higher than the threshold value from the ranking model. We found that we could predict the *exact* number of parents of about 70% of the new concepts using a normalized rank threshold of 0.4, and the correct number of parents ± 2 for about 87% of the new concepts. We also tried a local, concept-specific ranking threshold, based on the level (i.e distance from root) of the top ranked parents output by our ranker. However, this idea did not yield as good results, possibly due to the manual

curation of Wikipedia categories, or the presence of many edges from very general to highly specific nodes in the taxonomy. Hence we report all our results using this global normalized ranking threshold.

2.5 Evaluation

We tested ETF on two large-scale general-purpose taxonomies, Wikipedia Categories and WordNet, which we detail next.

2.5.1 Enhancing Wikipedia Category Taxonomy

To set up the experiment, we extract the entities and categories of the Wikipedia category hierarchy available via DBPedia [129], formatted as per the Simple Knowledge Organization System (SKOS). Wikipedia articles that are not category pages (containing lists of categories and sub-categories) are considered as entities. Relations between categories and entities are represented by the 'skos:broader' relationship. We then construct a DAG T, with the entities as leaf nodes and edges between entities and categories or among categories. This taxonomy exceeds 5 million concepts.

We now outline the procedure we adopt to generate the training and testing data for our approach. First, we use the hyperlink structure of Wikipedia to identify all noun phrases that are ambiguous, i.e., they have more than one concept associated with them as per the Wikipedia ontology, and which occur at least 10 times (i.e. $support \geq 10$). For example, Apple the company and apple the fruit are different entities that share the noun phrase "apple". We randomly sample 6000 of these ambiguous noun phrases. For each phrase, we randomly sample one sense (e.g., Apple, the company) and include it in our test set of new concepts to be inserted into the Wikipedia category taxonomy.

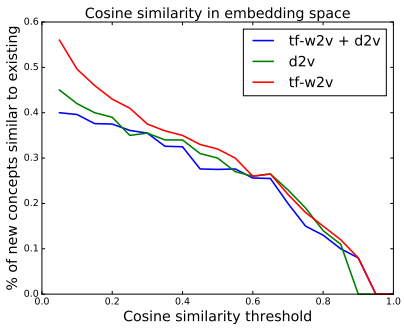


Figure 2.2: Investigating the quality of ETF embeddings in identifying new concepts from prevailing taxonomy entities

To ensure that the new concepts are independent of the existing ontology and do not have any presence within it, we also filter all Wikipedia pages that are linked to the sampled ambiguous entities. This process resulted in the removal of about 11% of the total Wikipedia pages. In the remaining 89% of the pages, we replace the occurrences of each concept or entity by its unique canonical identifier. This modified set of articles is used as textual input to train our embedding model (Section 2.4.2). The mentions of each new concept from the 11% of pages that were deleted are used to form contexts for the new concepts. To facilitate reproducibility, we provide (see `https://github.com/vnik18/Taxonomy`) the: (i) Wikipedia version we used; (ii) list of concepts, and; (iii) preprocessing scripts.

2.5.1.1 Evaluating Concept Representations

We begin by testing the quality of the aggregated vector embeddings of the new concepts, by checking how well separated they are from the existing entities' embeddings, i.e., all Wikipedia entities excluding the pages on the new concepts. The y-axis of Figure 2.2 shows the percentage of new concepts whose pairwise cosine similarity with the taxonomy entities (averaged over all entities) is greater than or equal to the cosine similarity threshold on the

x-axis. We evaluate the performance of the three kinds of embeddings from Section 2.4.2 in distinguishing the new concepts: (i) tf-weighted word2vec embeddings, (ii) doc2vec embeddings, and (iii) summing these two kinds of embeddings. Aggregating the doc2vec and tf-weighted word2vec embeddings gives us a 2-5% improvement in the quality of separability of the new concepts. From Figure 2.2, we observe that only about 28% of the novel test concepts have an average pairwise similarity of ≥ 0.4 with existing concepts using the aggregated embedding. In other words, more than 70% of the novel test concepts are well-separated from the prevailing entities in the embedding space at a cosine similarity threshold of 0.4.

2.5.1.2 Evaluating Our Ranking Model

Baselines: These include feature-based variants of our ETF ranker, and a state-of-the-art knowledge graph completion technique, *TransR*:

1. Random: It connects each new concept to a fixed number of randomly selected parents in T, equal to the average of the actual number of parents for the new concepts in ground truth (*ranker-rand*), i.e. 6 parents in case of Wikipedia.

2. Text similarity: Each new concept is linked to the top 6 parents with the highest text-based Jaccard similarity between the parents and the new concept (*ranker-textsim*).

3. Graph features: This baseline (*ranker-gr*) only makes use of the features based on the taxonomy DAG properties defined in Section 2.4.3.1, as input to the learning model.

4. Semantic features: We train two kinds of rankers using only semantic features (Section 2.4.3.2). The first, *ranker-sem-noPMI*, is trained on the textual features excluding PMI, and the second, *ranker-PMI*, is trained only on the PMI feature.

5. TransR: We use the TransR [182] based concept embeddings and the embedding of a single relationship type *parent*, to predict the top 6 parents of each new concept (*ranker-TransR*).

Performance Measures:

1. Precision@r: We report the precision of each approach, micro-averaged over each new concept from the testing set. This is the number of correctly identified parents in the top 'r' ranked parents, divided by the number of parents retrieved.

2. Recall@r: This is the (micro-averaged) number of correctly identified parents in the top 'r' ranked parents, divided by the correct number of parents of the new concept.

3. NDCG@r: We evaluate the performance of each approach by measuring the Normalized Discounted Cumulative Gain (NDCG). For a new concept x that has P parents $p_1, p_2, ..., p_P$, the discounted cumulative gain (DCG) at rank r and the ideal discounted cumulative gain (IDCG) at r are defined as:

$$DCG@r = \sum_{i=1}^{P} \frac{1}{\log_2(rank(p_i)+1)}; IDCG@r = \sum_{i=1}^{P} \frac{1}{\log_2(i+1)}$$

where rank(p_i) is the rank of parent p_i in the ranked list of parents deduced by the approach. $NDCG@r$ is the $DCG@r$ divided by the $IDCG@r$. Since NDCG is defined for a single concept or document, we report the micro-averaged NDCG which is the average of $NDCG@r$ over all test concepts.

Ranker Comparison Results: Table 2.1 displays the performance of our approach against baselines on inserting 6000 new concepts into the Wikipedia category taxonomy. We report the values of precision, recall and NDCG by considering the top r parents, where r varies

Table 2.1: Performance results on the Wikipedia taxonomy.

Approach	**NDCG@r**	**Precision@r**	**Recall@r**	**F1**
ranker-rand (r=6)	0.104	0.143	0.211	0.17
ranker-textsim (r=6)	0.328	0.293	0.381	0.331
ranker-gr	0.367	0.451	0.46	0.455
ranker-sem-noPMI	0.46	0.48	0.55	0.513
ranker-PMI	0.54	0.46	0.524	0.49
ranker-TransR (r=6)	0.612	0.69	0.61	0.647
ranker-ETF	0.73	**0.72**	0.67	0.7
ranker-ETF-PMI	**0.745**	0.68	**0.73**	**0.704**

for each new concept based on the number of parents whose rank scores are above the rank threshold. Trained on a combination of graph and semantic features excluding PMI, *ranker-ETF* is able to achieve an NDCG of 0.73 and F1-score of 0.7. This is a 6% improvement in F1-score and a 12% improvement in NDCG over using TransR-based embeddings in *ranker-TransR*. Including PMI, *ranker-ETF-PMI* achieves about 6% improvement in recall and 1.5% in NDCG, but a drop in precision. We reiterate here that we only use those ancestors of the nearest neighbor entities of the new concepts that are reachable to them in at most 3 hops, as prospective parents of the new concepts. This restriction allows us to cover about 83% of the true parents of the new concepts, eliminating the remaining 17% of the parents belonging to ground truth from the reach of our technique. That is, our approach has a performance upper bound equal to the proportion of coverage of true parents, i.e. 83%. Judging by the performance of the ranker learning only from graph-based features (*ranker-gr*) and that learning only from semantic features (*ranker-sem-noPMI*), we conclude that both kinds of features lead to the efficacy of our approach.

Table 2.2 shows the average number of parents our method was able to correctly detect ('Hits'), and the parents that were not in the ground truth but were reachable within k-hops

Table 2.2: Average percentage of top 'r' parents reachable from predicted parents of new concepts within k hops.

Hits	**1-hop**	**2-hop**	**3-hop**	**4-hop**	**5-hop**	**Misses**
0.664	0.684	0.729	0.738	0.746	0.746	0.2534

Table 2.3: Performance on the WordNet taxonomy for the SemEval 2016 Task 14, using the measures of Lemma Match, Wu&Palmer similarity (Wu&P), Recall and F1 score.

Approach	**Lemma Match**	**Wu&P**	**Recall**	**F1**
Random synset	0	0.227	**1**	0.37
FWFS	0.415	0.514	**1**	0.679
MSejrKU System 2	0.428	0.523	0.973	0.68
ranker-ETF	0.42	0.473	**1**	0.642
ranker-ETF-FWFS	**0.5**	**0.562**	**1**	**0.72**

from the true parents. For instance, they could be parents of the true parents, or parents of the parents of the true parents. We are able to correctly predict about 67% of the parents of the new concepts, and an additional 8% of reasonably good parents since they are reachable from the true parents in ≤ 4 hops. But they are slightly more generalized than needed. We completely miss 25.34% of the parents, of which 17% are missed due to the 83% coverage limitation of our algorithm.

Computational Cost and Feature Importance: Our framework inserts one new concept per minute on average. The bottlenecks are the random walk betweenness centrality, and the PMI API calls, as the remaining features can be computed in seconds. However, in principle, a fast batch PMI implementation [170] can replace individual API calls, so the main bottleneck is the random walk betweenness centrality feature.

The three most important features, based on individual performance, are information propagation, random walk betweenness centrality, and Katz similarity. For Katz similarity, we also experimented with a directed variant, and found that it was more correlated with information propagation than the undirected variant, with Pearson correlation values of 0.4 and -0.037, respectively. However, there was negligible change in performance using directed vs. undirected Katz in conjunction with the other features.

2.5.2 Evaluation on SemEval Task Benchmark

We evaluate ETF on the WordNet 3.0 onotology [79], on the dataset of new concepts provided as part of the SemEval 2016 Task 14 [144]. It was specifically constructed to span a wide range of domains, including online resources such as Wiktionary and other glossary websites. This dataset contains 1000 concepts, split into a training set of 400 and a testing set of 600 concepts, which are either nouns or verbs. The task consists of choosing one of two operations for each new concept; (i) attach, where the novel concept needs to be added as a new synset in WordNet, and (ii) merge, where the novel concept needs to be added into an existing synset. For each concept, a textual definition of a few sentences and a single correct parent for it has been provided as part of the task. ETF chooses as the parent for each new concept the top ranked ancestor by our ranker. The design of our algorithm does not permit us to 'merge', we only perform the 'attach' operation for every unseen concept.

2.5.2.1 Experimental Setup

At the outset, we establish a DAG of the WordNet taxonomy, consisting of all the existing entities or leaves, connected to their respective categories via hypernym-hyponym relationships. Each WordNet synset corresponds to a single DAG node. We use the definition of the test concepts provided by Task 14 as their context. We infer the embedding for each

new concept as in Section 2.4.2, from the doc2vec model trained on Wikipedia articles, after making sure that any article pages on the new concepts, and pages that link to them have been removed. We then find the existing Wikipedia entities that are most similar to the new concept in the embedding space. However, one issue that arises here is that these nearest neighbor entities and their hypernyms are part of the Wikipedia category taxonomy, whereas we want to link the new concept to its most likely WordNet hypernyms. Hence, we employ the YAGO ontology [304] to link the Wikipedia neighbor instances to the corresponding instances at the leaf level of the WordNet hierarchy. However, all Wikipedia neighbor entities may not have corresponding WordNet counterparts. We eliminate all those nearest neighbor entities from consideration which do not map to a WordNet instance. We finally compute the graph-based and semantic features as in Section 2.4.3, from the constructed WordNet DAG and embedding model, to rank the potential hypernyms of the new concepts.

2.5.2.2 Baselines and Performance Measures

To evaluate our approach, we use as baselines those used in Task 14, and the winning system of the task, *MSejrKU System 2* [279]:

1. Random synset: It attaches the new concept to a random WordNet synset, with the same part-of-speech as itself.

2. First-Word-First-Sense (FWFS): It links the new concept to the first word in its definition with the same part-of-speech as itself, stemming from the grouping of glosses in WordNet.

3. MSejrKU System 2: This was the system that won the SemEval Task 14.

We assess our method using the measures defined in Task 14:

1. Accuracy (Wu&Palmer Similarity): Wu and Palmer [368] defined the semantic similarity between the predicted parent x_1 and the true parent x_2 as: $Wu\&P(x_1,x_2) = \frac{2*depth_{LCA}}{depth_{x_1}+depth_{x_2}}$ where LCA is the Least Common Ancestor of x_1 and x_2, and $depth_{x_i}$ refers to the depth of x_i in the WordNet hierarchy.

2. Lemma Match: It measures the proportion of test concepts for which a method selects a synset with at least one word in common with the correct synset for that concept.

3. Recall: This is defined as the percentage of test concepts for which an output ancestor was identified by an algorithm.

4. F1 score: Harmonic mean of the Wu&Palmer score and recall.

2.5.2.3 Results

As mentioned earlier, though the task permits two operations of 'attach' and 'merge', we only perform the 'attach' operation. Just 'attaching' every new concept to its own independent synset causes the ceiling (upper bound) on the achievable F1-score to be about 0.989, with precision 0.98 and recall 1. Table 2.3 exhibits the performance of ETF on the SemEval task on WordNet. We evaluate two versions of our approach, (i) a ranker trained on the same set of features as described in Section 2.4.3 (*ranker-ETF*), and (ii) the same ranker using an additional binary feature of the FWFS property (*ranker-ETF-FWFS*): if a prospective parent of an unseen concept is the first word of its definition with the same part of speech as itself, the value of this feature is 1, otherwise it is 0. We observe that our ranker without the FWFS feature gives an F1 score of 0.642, however with the FWFS-based binary feature, ETF's F1 score outperforms the best system of this task by 0.04. This is particularly

impressive given that very little improvement compared to FWFS (0.001 in F1 score) has been recorded on this benchmark in the past.

We note that FWFS is a strong feature for ranking parents in the WordNet taxonomy, since the organization of word glosses in WordNet ensures the presence of the word expressing the ancestor concept early in its gloss, rather than later. However this feature is not easy to generalize for other taxonomies.

2.5.3 Emergent Domain Concepts

We perform a case study to examine how ETF adds newly emerging concepts into the Wikipedia category hierarchy, focusing on the domains of crisis response and medicine (Table 2.4). We select a set of concepts, and eliminate them and all Wikipedia pages linking to them from consideration. To keep the evaluation process fair, we use as text input to our algorithm, the versions of the Wikipedia article pages on the concepts that existed when the concept was first added. Column 2 of Table 2.4 shows manually assigned parents for the original versions of the new concept pages. Column 3 shows the manually allocated categories on the present-day Wikipedia versions of the new concept pages, and column 4 shows the top ranked parents predicted by our approach. In column 4, we show in bold the predicted parents that match with those in column 3, and italicize the parents that are good predictions but are not in column 3. For both domains, most predicted parents are quite good without being overly general, and overlap with many of the manually assigned parents. For the crisis events (rows 1, 2, 3), ETF correctly identifies their year of occurrence and a good number of affected areas, with just a preliminary amount of text input. With respect to the medical domain, our method can seemingly discriminate between *Avian Influenza* and *Swine Influenza* (rows 4, 5, 6). We are also able to propose accurate categories for new

concepts that have not been manually assigned: *Subtypes of Influenza A virus* for *Avian influenza*, and *2009 flu pandemic* for *Swine Influenza*.

Overall, the parents predicted by ETF for all the emergent concepts are better than the human-assigned categories to them at that point in time. The results indicate that our algorithm can accurately organize new concepts across varied domains.

2.5.4 Quora Q&A Categorization

As a use case for ETF, we consider mapping questions and answers (Q&A) from Quora to appropriate Wikipedia categories (last 4 rows of Table 2.4). These have a different style of writing than the more definitional style of Wikipedia articles. Moreover, a single question and answer can span many topics. One feature of Quora is that many of the existing tags assigned to questions directly map to existing categories and entities from Wikipedia. We selected 384 questions from a diverse selection of categories, and processed the first paragraph of the question and top answer using our framework, treating this text as a definition for a new concept. On average each question had been tagged with 2 concepts from Wikipedia. Some of these concepts were entities, and some of them were categories. For the ground truth, we consider the union of the tagged categories and parent categories of the tagged entities. Overall, this resulted in an average of 8 parents per Q&A. ETF achieved an NDCG of 0.445 and F1 score of 0.523, and was, similar to previous results, an improvement (NDCG increase of 5.5%, and F1 increase of 8.7%) over the other baselines. The scores on this test were lower across all aproaches, compared to the Wikipedia-based case study. We believe the primary reasons for this drop are that the ground truth is much sparser, and also that this task is more difficult than that of Section 2.5.1, as there is more concept mixing in the Quora Q&A compared to the focused writing on Wikipedia.

Table 2.4: Case Study: Performance of ETF on adding emergent domain concepts into the Wikipedia category taxonomy (first six rows). Quora Evaluation: example questions and comparison of manually assigned vs. predicted categories (last four rows).

New concept	**Initial parents**	**Current manually assigned parents**	**Predicted parents by ETF**
Tropical Storm Erika (2015)	Current events from August 2015	2015 Atlantic hurricane season, 2015 in the Caribbean, Natural disasters in Dominica, Hurricanes in Desirade, Hurricanes in Dominica, Hurricanes in Florida, Hurricanes in Guadeloupe, Hurricanes in Haiti, Hurricanes in the Bahamas, Hurricanes in Puerto Rico, Hurricanes in the Dominican Republic, Hurricanes in the Leeward Islands, Atlantic tropical storms	**Hurricanes in Florida**, *Natural disasters in Florida, Tropical cyclones*, Articles which contain graphical timelines, *Atlantic hurricanes*, **Hurricanes in Guadeloupe, Hurricanes in Dominica**, Tropical cyclone seasons, *Typhoons*, **2015 Atlantic hurricane season**, Tropical cyclones by strength, *Natural disasters in Guadeloupe*, Tropical cyclones by basin
Illapel Earth-quake, Chile (2015)	Current events, 2015 earthquakes	2015 earthquakes, 2015 tsunamis, 2015 in Chile, Megathrust earthquakes in Chile, Tsunamis in Chile, September 2015 events	**2015 in Chile**, *Earthquakes in Chile*, **2015 tsunamis, Tsunamis in Chile, Megathrust earthquakes in Chile, 2015 earthquakes**
Shootings at Parliament Hill, Ottawa (2014)	Current events, 39th Canadian Parliament	Attacks in 2014, 21st century in Ottawa, 41st Canadian Parliament, Terrorist incidents in Canada in 2014, Attacks on legislatures, Crime in Ontario, Deaths by firearm in Canada, October 2014 events, Parliament of Canada, Spree shootings in Canada, Political controversies in Canada, ISIL terrorist incidents in Canada, 2014 in Ontario	**2014 in Ontario**, Military history of Ontario, Ottawa, **Parliament of Canada**, Monuments and memorials in Ottawa, History of Ottawa, *Terrorist incidents in 2014*, **Crime in Ontario**, **Attacks in 2014**, Religion in Canada, **Political controversies in Canada, 21st century in Ottawa**, *Terrorist incidents in Canada*, **Spree shootings in Canada**
Avian Influenza	Current events, Influenza	Animal virology, Bird diseases, Avian influenza, Poultry diseases, Agricultural health and safety	*Subtypes of Influenza A virus*, **Bird diseases**, *Animal diseases*, **Poultry diseases**, *Viral diseases*
Influenza A virus subtype H7N9	Subtypes of Influenza A virus	Subtypes of Influenza A virus, 2013 health disasters, Health disasters in China, 2013 disasters in China	**2013 health disasters, Health disasters in China, 2013 disasters in China**, *Bird diseases*
Swine influenza	Medicine stubs, Pandemics	Animal virology, Health disasters, Swine diseases, Influenza, Pandemics	*2009 flu pandemic, Influenza A virus subtype H1N1*, **Influenza**, *Influenza pandemics*, **Health disasters**
Does over-working help Japan's economy?	–	Applied sciences, Social sciences, Economy of Japan, Processes, Economy of Asia, Economies by country, Economics, Japan	*Social impact*, **Economy of Japan**, **Economies by country**, Retailing by country, **Economy of Asia**, *Economies by region*, *Economywide country studies*
Why do lightnings have a branch structure?	–	Space plasmas, Atmospheric electricity, Lightning, Storm, Electrical phenomena, Weather hazards, Electric arcs, Electrical breakdown	*Meteorological phenomena*, Electric power transmission, Electric power, *Physical phenomena*, **Weather hazards, Electrical breakdown, Storm**

Table 2.4 Continued: Case Study: Performance of ETF on adding emergent domain concepts into the Wikipedia category taxonomy (first six rows). Quora Evaluation: example questions and comparison of manually assigned vs. predicted categories (last four rows).

New concept	**Initial parents**	**Current manually assigned parents**	**Predicted parents by ETF**
How true is the belief that natural things are good?	–	Main topic classifications, Science, Science technology engineering and mathematics, Nature, Physical universe	**Nature**, *Philosophical theories*, Biological interactions, **Physical universe**, History of science by discipline
What did Han Xin do wrong to be killed by Liu Bang?	–	China, Asian royal families, Chinese-speaking countries and territories, Han dynasty, History of Ancient China, History of China, Dynasties in Chinese history, Iron Age Asia	**Dynasties in Chinese history**, *Han dynasty people*, Qin Dynasty, **Asian royal families**, History of Asia by country, **History of Ancient China**, *People by Imperial Chinese dynasty, Histories of cities in China*

2.6 Conclusion

In this work, we propose a solution to the problem of automated taxonomy enrichment, where we insert new concepts at appropriate positions into an existing taxonomy. Our proposed approach ETF learns a high-dimensional vector embedding via a generated context of terms, for each existing and new concept. We then predict the potential parents of the new concepts from the ancestors of their nearest neighbors, by ranking them based on semantic and graph features. We evaluated ETF on the large knowledge bases of Wikipedia and WordNet, and could outperform other baselines.

ETF has the potential to be applicable under variations in text sources (e.g. short, informal social media text) and types of taxonomies (e.g. enhancing taxonomies belonging to specific domains). It allows for easy parallelization and can be distributed for scalability. Further, all features used by ETF can be computed in a few seconds except for the random walk betweenness centrality feature.

Chapter 3: BOLT-K: Bootstrapping Ontology Learning via Transfer of Knowledge

In the previous Chapter, we investigated the problem of augmenting existing contextual knowledge sources with newly emerging additional information. In contrast, in Chapter 3 we now study how to create entire structural schemas for unfamiliar, domain-specific ideas or events. This problem poses the following challenges: (i) detecting relevant, often previously unknown concepts associated with the new domain; and (ii) learning ontological, semantically accurate relationships among the new concepts, despite having severely limited annotated data. We propose a novel LSTM-based framework with attentive pooling, BOLT-K, to learn an ontology for a target subject or domain. We bootstrap our ontology learning approach by adapting and transferring knowledge from an existing, functionally related source domain. We augment the inadequate labeled data available for the target domain with various strategies to minimize human expertise during model development and training. BOLT-K first employs semantic and graphical features to recognize the entity or concept pairs likely to be related to each other, and filters out spurious concept combinations. It is then jointly trained on knowledge from the target and source domains to learn relationships among the target concepts. The target concepts and their corresponding relationships are subsequently used to construct an ontology. We also examine the potential of BOLT-K in detecting the presence of novel kinds of relationships that were unseen during training.

3.1 Introduction

Domain-specific ontologies are valuable resources that formally model the conceptual vocabulary of a given domain. Building accurate ontologies from trustworthy sources [198, 253, 343, 344, 354] with sufficient coverage of concepts and relationships among them is time-consuming and labor-intensive. A number of approaches to automate this process have been proposed. For instance, building ontologies based on statistical, linguistic and graphical features [226, 348, 387]; enriching existing ontologies with domain-specific information [339]; and embedding-based methods to complete knowledge graphs [27, 138, 182, 229, 328, 329, 360, 376, 392, 393].

Nevertheless, they only utilize textual corpora and other accompanying information from the specific domain under consideration. They do not leverage the abundant, hierarchically structured knowledge that might be available in functionally and/or semantically *similar* or *related* subjects or domains. For instance, there is no semantically coherent ontology associated with the recently surfaced human disease of *Zika fever*, knowledge of which is evolving to-date. It will therefore be highly useful to take advantage of its connections to similar vector-borne diseases like *Dengue* or *Malaria*, for which well organized and annotated information from domain experts is available. Further, there are multiple challenges associated with building comprehensive ontologies for e-commerce product platforms [71, 174]. Numerous closely related product catalogs often have incomplete or incorrectly labeled attributes. Using annotations from semantically similar product listings (such as products from the same category) can help embellish existing listings with missing attributes, as well as detect errors in the labeled attributes.

A crucial task in constructing ontologies is learning hierarchical relationships among multiple concepts from unstructured text. This task requires large amounts of annotated

training data associated with ontological concepts and their corresponding relationships. This process is time-consuming, expensive, and necessitates a significant amount of expert knowledge to categorize associations in niche domains that a layperson is unlikely to know about. To resolve this issue, Mintz et al [215] heuristically aligned texts with knowledge graphs via distant supervision to automatically generate training examples. Lin et al [183] applied distant supervision with sentence-level selective attention, while Zeng et al [392] coupled it with multi-instance learning to learn relationships. However such efforts either do not address the problem of the highly imbalanced occurrence of different relationships [260], or require a sizable number of sentences connecting related entities. Another popular strategy to address the issue of insufficient labeled data is *data augmentation*. This technique artificially expands labeled training sets by generating new data points or transforming the existing ones, such that the class label properties are preserved [134, 195, 257].

In this chapter, we propose a framework *BOLT-K* – Bootstrapping Ontology Learning via Transfer of Knowledge. It uses a long short term memory (LSTM) neural network with attentive pooling, to learn an ontology hierarchy for a given *target* domain. We first obtain all the concepts that constitute the target ontology. We subsequently devise an approach based on semantic and topological attributes to identify the concept pairs likely to be connected by a relationship, and eliminate the remaining spurious combinations. To address the issue of limited relationship-labeled training data, we *bootstrap* the learning task by utilizing publicly available textual corpora and ontological information from a functionally similar *source* domain. We also employ data augmentation techniques to generate additional training examples for the target domain. We train our model jointly on the target and source hierarchy information, by sharing the hidden feature representations and appropriate model parameters among them. Finally, we predict ontological relationships between the concepts of the target

domain, to construct an ontology for it. We extensively evaluate our framework on several real-world datasets, highlighting the transferability of concepts across comparable subjects. We show that BOLT-K can significantly improve the quality of learned ontologies over state-of-the-art baselines, when bootstrapped with relevant knowledge from a similar subject or domain. To summarize, the key contributions of our work are:

- We develop a flexible and generalizable framework, BOLT-K, to automatically learn ontologies for contemporary novel or emergent sub-domains of rapidly evolving fields such as bio-medicine, epidemiology, e-commerce and crisis response.
- We propose to transfer existing knowledge from functionally similar domains, and augment the insufficient labeled target training data. This significantly lessens the need for manual expertise during model development and training.
- We extensively evaluate BOLT-K on real-world datasets for various sub-domains within bio-medicine and product graphs. We also show BOLT-K's capability in detecting novel types of relationships that were unseen during training.

3.2 Related Work

The following lines of research are related to our work: (i) augmentation of textual training data; (ii) learning ontological relationships; and (iii) transfer learning for natural language processing tasks.

Training data augmentation: This is a common and successful technique in the image processing literature [72,257,272], and is slowly gaining popularity in NLP applications [134, 195,257]. A caveat though, is that supplementing text data via transformations, interpolations or affine perturbations is not as straightforward as performing these enhancements for

images. Kafle et al [146] proposed two methods of data augmentation for visual question answering: (i) using semantic segmentation annotations with labels to synthesize certain kinds of questions; and (ii) training a stacked LSTM model to generate questions about images. A common technique of augmenting text is to replace words or phrases with their synonyms from a thesaurus [396], or with appropriate n-grams from a language model [257]. Another technique is to translate sentences into a second language, and then translate them back into the original language to obtain a slight variation of the original sentence [75, 271, 366]. Progress has also been made in developing generative models based on variational autoencoders for sentence generation [53, 130, 286]. In this work, we employ a combination of text substitutions for data augmentation, as specified in Section 3.4.2.

Relation Extraction: Deep learning models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs) and graph-RNNs have been successful in automatically learning features for extracting semantic relations between a pair of entities [243, 276, 370, 392]. A number of approaches have also employed attention mechanisms in conjunction with RNNs for relation extraction and classification [183, 401]. Another way of solving this problem is by formulating it as a link prediction problem in knowledge graphs [27, 138, 182, 229, 328, 360, 376]. Since our method addresses relation extraction at the level of a sentence or a group of sentences, we find that a bidirectional LSTM with attention-based pooling works well for our purpose. Similar to our work, there have been efforts to extract relations with an insufficient number of labeled examples per relationship type. Yuan et al [390] formulated this as a one-shot classification problem and solve it using a convolutional siamese neural network. Levy et al [177] turned it into a reading comprehension problem by associating multiple textual questions with each relation type and learning answers for them.

Transfer learning: Zhang et al [395] have surveyed various cross-dataset transfer learning techniques. This includes the kind of knowledge transfer paradigm that our work addresses, i.e. minimizing the generalization error in the target domain with the help of training instances from two disjoint source and target domains. Transfer learning has been highly useful in low-resource domains such as bio-medicine [154, 211]. Ganin et al [91] introduced a representation learning approach for domain adaptation by adding a gradient reversal layer in a feed forward neural network. They trained their model on a document sentiment classification task using labeled data from a source domain and unlabeled data from a target domain. Long et al [192] presented another approach that learns an unsupervised residual function to adapt classifiers from a source domain to a target domain. Similar to our work, prior efforts addressed the sharing of structural parameters across multiple task domains [6]. Yang et al [381] developed a framework for three different types of transfer learning paradigms in hierarchical RNNs for a sequence tagging task; namely cross-domain, cross-application and cross-lingual transfer. Knowledge transfer has also been used to inform inter-related NLP tasks such as named entity recognition, part-of-speech tagging, chunking and word segmentation [57, 242].

3.3 Problem Formulation

We propose to learn an ontology for a *target* domain for which labeled information is very limited, by transferring potentially useful semantic and ontological knowledge from a distinct but related *source* domain. We refer to each element of the ontologies associated with these domains as an *entity* or a *concept*. The inputs to our problem are:

1. A knowledge ontology S consisting of entities, categories and labeled relationships S_R among them, for a source domain.

2. A small number of concept pairs that will be part of the ontology T for the target domain. These have been labeled with their respective relationship type from a set of target relationship types T_R. Since the target is an emergent domain with little to no labeled information available, we only require at least one labeled concept pair per relation type in T_R. Note that though S and T are from related domains, there may or may not be relation types common to S and T. This is not a requirement for BOLT-K.

3. Corpora of text documents whose sentences contain occurrences of the concepts in S and concepts associated with the target domain, which are to be part of T. These documents can either be expert-authored such as research papers, or can be from public resources such as news articles or encyclopedias. We utilize the text corpora associated with the target domain to extract the set of concepts to be inserted into T, as explained in Section 3.4.1.

We extract the sentences containing co-occurrences of the concept pairs linked by relationship types in S_R as part of training data from the source. However, it is unlikely for a whole lot of information to be available about the emergent, target domain concepts in the corpora. Thus as part of the target training data, we restrict BOLT-K to use at least one and at most five labeled instances per relationship type in T_R. These would contain co-occurrences of a small number of target concept pairs. We employ data augmentation techniques to enhance this limited amount of training data (Section 3.4.2). Our BOLT-K approach thus makes use of (i) the abundant labeled relationship information from S and; (ii) the minimal amount of labeled relation information from T, to learn relationships between various pairs of concepts for the target domain.

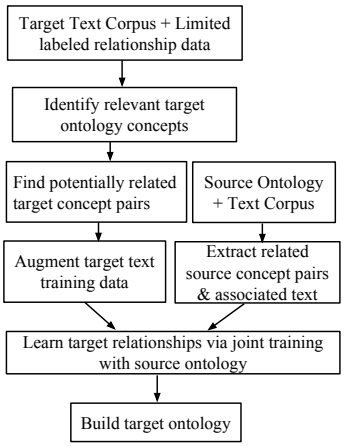


Figure 3.1: Pipeline of our BOLT-K framework

3.4 The BOLT-K Framework

In this section we describe our proposed BOLT-K approach (Figure 3.1). Among the set of concepts that are to be part of the target ontology T, we first identify the pairs of concepts that are likely to be related to each other in T, and filter out the remaining spurious pairs. Next, we augment the limited number of available training instances that are labeled with the target relationships from T_R. We then use this dataset along with the annotated information from S to learn the relations between the remaining target concepts in T. The identified concept pairs for T and the relationship types between them that have been predicted by our model can be used to construct a complete ontology for the target domain T.

3.4.1 Filtering Unrelated Target Concept Pairs

This is the first step of our algorithmic pipeline. We acquire the entity and category concepts from the given source ontology structure S. We assume for the purpose of simplification, that a list of the concepts that need to be a part of the target ontology T has been

provided to us by a domain expert. Alternatively, we can also use emerging entity extraction algorithms [68, 77] to locate and extract concepts from the target text document corpus. This corpus can consist of news reports, laboratory records or research articles authored by domain experts, related to the target topic. It is highly inefficient (quadratic complexity) and unnecessary to consider all possible pairs of target concepts while learning relationships between them. Hence, our next step is to identify the concept pairs likely to be linked by a relationship and filter out the remaining pairs. Our empirical investigations (Figure 3.3) show that using only semantic information is insufficient to capture the likelihood of a relationship between pairs of target concepts. Hence, we also incorporate structural information by modeling the target concepts as a graph.

We construct a weighted concept graph T_G, whose nodes consist of the concepts that are to be a part of the target ontology T. We link two nodes by an edge if they have co-occurred together at least once in a document in the target text corpus, within W words of each other. Inspecting a sample of documents in our corpora showed that related concepts often co-occur not in the same sentence, but in consecutive sentences. Since the length of an average sentence is about 10-12 words, we use $W = 25$ to indicate adjacent sentences. The weight of an edge is given by the pointwise mutual information (PMI) between the two concept nodes of the edge:

$$PMI(c_1, c_2) = log\left(\frac{num(c_1,c_2)}{num(c_1) \cdot num(c_2)}\right)$$

$num(c)$ (or $num(c_1, c_2)$) is defined as the number of occurrences of a particular concept (or co-occurrences of a pair of concepts within W words of each other) in the target text corpus.

The PMI metric semantically gives us a good sense of the possibly related target concepts. Nevertheless, it yields a number of false positives which we seek to eliminate, by utilizing

additional topological properties of the concept graph. We examine if the local neighborhood of a concept is a sufficient indicator of its potential relationships. However, we empirically find that local structural information is insufficient to account for concept relationships (Figure 3.3). Therefore, we employ a more global measure for this purpose that takes into account the overall concept graph topology, namely edge-based random walk betweenness centrality [228] (also called current-flow betweenness centrality [29]). For each edge in the weighted graph T_G, it measures approximately how often a node is traversed by a random walker going from any node in the network to another. If a concept c_1 appears often on random walks from concept c_2, then it is likely to be related to c_2. Finally, we learn a global threshold based on the betweenness centrality values, and consider only those target concept pairs as potentially related if their edge betweenness centrality value is above the threshold.

We now have the set of source concept pairs that are related via labels in S_R to each other from the source ontology S. We have also obtained the set of target concept pairs that are likely to be related to each other and which will form the target ontology T. We now generate a training dataset for both the source and target domains, which will serve as input to the next step of BOLT-K's pipeline, i.e. learning the relationship between the pairs of target concepts. From the source and target document corpora, we extract sentences or blocks of sentences in which potentially related concepts co-occur within W words of each other. Henceforth, for ease of understanding, we call a group of W consecutive words a *sentence*, even though they may physically span more than one sentence of text. We mark the occurrences of the related concepts or entities in each sentence. As mentioned earlier, the target dataset may not have an abundant amount of labeled data available. To account for this, we limit BOLT-K to use at most 5 labeled concept pairs (and hence at most 5 sentences containing them) per target relationship type for training.

3.4.2 Data Augmentation for the Target Domain

A crucial requirement of building a predictive model is the availability of a sufficient amount of labeled training data for the relationship types of the target domain. Hence, to control generalization error and avoid overfitting to the limited training data available, we adopt the technique of *training data augmentation*. It artificially enhances labeled training datasets by transforming the available data items such that the class label properties are preserved. This lends a major advantage to our approach, i.e. the ability to transfer properties from one taxonomic hierarchy to another at no additional annotation cost.

Some effective methods of performing text data augmentation with minimal human effort could be to create sentence paraphrases, or to substitute specific words or phrases with likely candidate words (e.g. synonyms). In our work, we augment the relationship-annotated target training sentences by replacing chosen words in them. We want to do this as diversely as possible, such that the syntactic and semantic equivalence between the original and altered sentence is maintained. Inspired by Ratner et al [257], we first identify the noun, verb and adjective terms in the target training sentences using the StanfordCoreNLP part-of-speech tagger [205]. We filter out the terms that do not occur above a learned frequency threshold. Out of these selected terms, we then iteratively sample a term occurring to the left, in between, and to the right of a pair of entities or concepts in each target sentence for substitution. We do not replace more than two terms in a single sentence at a time, to preserve the meaning and grammatical correctness of the modified sentence. We replace the chosen terms in the following three ways.

The first is by constructing an n-gram language model [257]. It is built by recording the frequencies of n-gram occurrences in the source and target corpora, filtering out the less frequent n-grams, and applying Laplace smoothing to the n-gram counts. This model

samples words conditioned on the words preceding them. It identifies the n-gram n_x preceding the word or phrase x to be replaced. It then finds from the corpora a list of terms l_x, following n_x, sorted in descending order based on frequency. It finally replaces x with the term at index i in l_x. i is picked based on a geometric distribution $P[i] \sim p^i$, where a more frequent term has a higher probability of being chosen as a substitute [397]. The value of p is fixed at 0.5. The n-gram model falls back to using bigrams (or unigrams), in case the required trigram (or bigram) was filtered out.

For our second technique, we replace the chosen words in each sentence with one of their synonyms from their synset gloss in WordNet [79]. Since synonyms in a gloss are ranked according to how frequently they are observed in natural language, we use a similar geometric distribution as mentioned above to pick a synonym substitute. For our third substitution strategy, we use a pre-trained word2vec [213] word embedding model induced on texts from PubMed, PMC and the English Wikipedia [217]. After obtaining vector representations for each of the terms to be replaced, we substitute them with the terms most similar to their vector representations in the embedding space. We present a comparative evaluation of the three augmentation strategies in Table 3.3. Once we enhance the labeled dataset for the target domain, we use this data in combination with the labeled examples from the source domain to train the core model of BOLT-K, as described in Section 3.4.3.

3.4.3 BOLT-K Core Model

We now introduce the core model architecture of BOLT-K. It utilizes human-annotated knowledge on ontological concept relationships from a source domain, and minimally labeled and artificially augmented data from a functionally similar target domain, to learn a hierarchical ontology for the target domain. We build an LSTM-based model with attentive

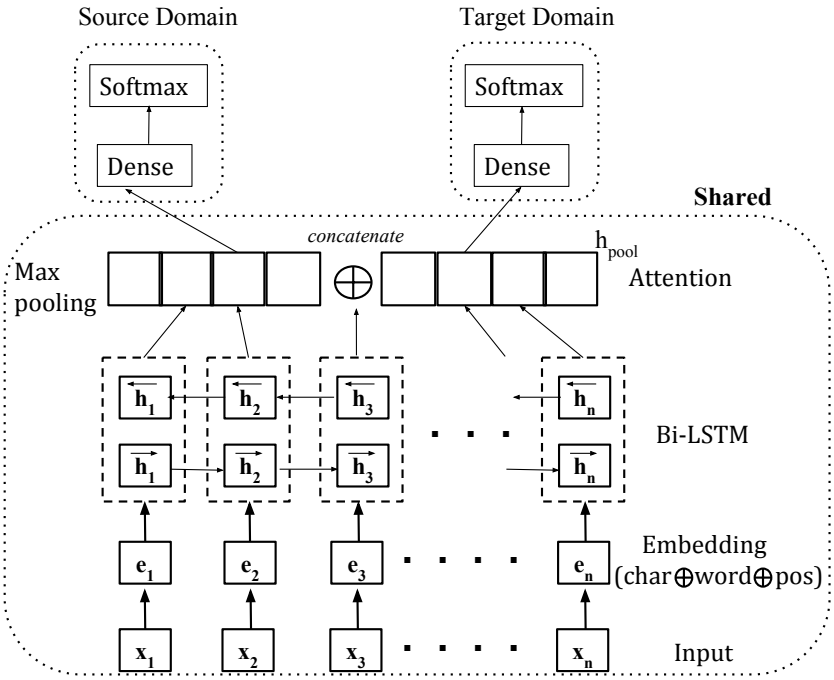


Figure 3.2: Proposed BOLT-K model architecture

pooling to learn an ontology for the target concepts. We transfer ontological relationship knowledge from the source to the target domain in this model, by sharing the hidden layer representation and some of the model parameters between the two domains. We also combine the objective functions of both domains for effective training. Such models have been used in the literature for transfer learning in various applications [381, 395].

3.4.3.1 Base Model Components

Figure 3.2 presents an overview of the core model of our proposed BOLT-K framework. It consists of two parts: a *base* model architecture which is shared among both the source and target domains, followed by domain-specific neural network layers. We first describe each of the base model components in detail.

The input to our model is a set of sentences obtained from the source or target text corpora. Each sentence consists of n words $[x_1, x_2, ..., x_n]$. These words include a pair of ontology entities linked by a relation label. The next layer of our model is an embedding layer, which represents every input word x_i as an embedding d_i. d_i is formed by concatenating the character-level representation of x_i with its word embedding from a pre-trained word2vec model [217]. Words which lack embeddings in this model are given a random representation. We obtain the character-level representation of each word using a CNN. CNNs have been shown to encode useful morphological information like word prefixes and suffixes from the characters of a word [55, 199, 275]. Our CNN model consists of a convolution layer followed by dropout and max pooling. Its input is randomly initialized for each character of a given word x_i, and its output is a character-level representation of x_i. We form a final embedding e_i by appending two additional position indicators to each d_i. These are the normalized word-distances of word x_i from both the related concepts in their respective sentence. The embedding matrix $[e_1, e_2, ..., e_n]$ is updated during model training and serves as input to the next layer, i.e. the bidirectional LSTM.

Bidirectional LSTM [106] based models are used for a variety of sequence modeling tasks where it is often beneficial to utilize both the past and future context. These networks extend the traditional uni-directional LSTM [123] units that only consider past sequential information, by accounting for temporal context information from future time steps. At the core of the LSTM unit is a memory cell controlled by three sigmoidal gates: the input gate i_t deciding whether the unit retains its current input x_t or not, the forget gate f_t to enable the unit to forget its previous memory context c_{t-1}, and the output gate o_t controlling the context transferred to the hidden state h_t. The recurrences for the LSTM are defined as:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$h_t = o_t * \tanh(c_t)$$

where σ is the sigmoid function, tanh is the hyperbolic tangent function and $*$ represents the product with the gate value. W, U and b are matrices of network parameters to be trained. As shown in Figure 3.2, the Bi-LSTM layer combines its forward ($\overrightarrow{h_t}$) and backward ($\overleftarrow{h_t}$) sequence contexts using the concatenation operation ($\oplus$). The output h_t of this layer at time step t is given by: $h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t}$

The Bi-LSTM layer generates a sequence of word-level representations $[h_1, h_2, ..., h_n]$ utilizing past and future context, where $h_{j,t}$ denotes the j-th element of h_t. We perform one-dimensional max pooling to obtain a fixed length vector from the Bi-LSTM output.

$$m_{pool,j} = \max_{1 \leq t \leq n} [h_{j,t}]$$

The max pooling operation assumes that the important and relevant latent semantic features in the sentence are present at the positions [171] containing the maximum value of h_t. However, this might not always be the case. Hence, to focus on words that are crucial in predicting the relationship between entities which might lie anywhere in a sentence, and to give less importance to irrelevant information in the sentences, we utilize an *attention* mechanism. It highlights the important tokens in a given input sequence, which are responsible for performing feature selection for the model as well as for its predictions. Inspired by Yang et al [382], we introduce a word-level attention layer to capture the similarity of a word token with respect to its neighboring context tokens in an input sequence. It assigns weights to the hidden outputs h_t from the Bi-LSTM layer as follows:

$$z_{j,t} = \tanh(W_z h_{j,t} + b_z)$$

$$\alpha_{j,t} = \frac{exp(W_a^T z_{j,t})}{\sum_t exp(W_a^T z_{j,t})}$$

$$att_{pool,j} = \sum_t \alpha_{j,t} h_{j,t}$$

Here, W_z and b_z are the weight matrix and bias vector respectively associated with the hidden state h_t from the Bi-LSTM layer, and $h_{j,t}$ represents the j-th sentence. Their non-linear transformation yields z_t, for which W_a is the corresponding weight matrix. $\alpha_{j,t}$ are the normalized attention weights representing token importances from a softmax function at time step t. $att_{pool,j}$ is the attention-focused hidden state representation, given by the linear combination of the Bi-LSTM output h_t and the attention weights.

The final output of the base model $h_{pool,j}$ is the concatenation ($\oplus$) of the outputs of the max pooling layer and the attention layer.

$$h_{pool,j} = m_{pool,j} \oplus att_{pool,j}$$

3.4.3.2 Knowledge Transfer Components

All the layers in the base model described in Section 3.4.3.1, namely the input, embedding, bidirectional LSTM, attention and pooling layers are shared and informed by training data from both the source and target domains. The output from the base model or the shared block in Figure 3.2 serves as input to two separate fully connected dense layers, followed by two softmax layers, one each for the source and target domains. The softmax function predicts the relationship type between a pair of concepts in an input sentence.

3.4.3.3 Training

We next outline how we transfer information from the source to the target domain by training our model jointly for both domains. We adopt a training procedure similar to that

described by Yang et al. [381]. At each iteration, we sample one of the domains from the source and target based on a binomial distribution, where the binomial probability is a tuned parameter. We then optimize the objective function of the chosen domain by training on a sampled batch of labeled instances. The parameters of the shared block are thus updated due to training inputs from both domains, while the fully connected layers are only affected by their corresponding domain. We repeat this procedure until convergence, with early stopping based on the target domain performance.

We used 200-dimensional LSTM units with L2 regularization in our framework. We optimized the cross-entropy error between the true and predicted labels using Adam [160] with gradient clipping. We set the initial learning rate to 0.001 with a decay of 0.05. Dropout [298] was applied to the Bi-LSTM and pooling layers with a probability of 0.5.

3.5 Evaluation

3.5.1 Data Collection

We used the Open Biological and Biomedical Ontology (OBO) Foundry [292] and the National Center for Biomedical Ontology portal [222] which are collaborative repositories of science based ontologies, to obtain a family of ontologies related to the bio-medical domain. Each ontology has been curated with the important and relevant concepts associated with its specific sub-domain or subject, as well as the relationship types among the concepts. We also created and used commercial product ontologies from the open-source Web Data Commons project [251]. It contains structured data extracted from the web on different topics. Table 3.1 shows the statistics of various ontologies we have tested our BOLT-K framework on. These include for each ontology the number of concepts present in it, the number of related concept pairs, the total number of relationship types, and the median

number of sentences available for a single relationship type. The first six rows are based on various human diseases and disorders. The next two rows show ontologies of flowering plants (*Angiosperms*) and non-flowering plants (*Gymnosperms*). The final three rows are based on three popular kinds of commercial products that are frequently manufactured, bought and sold, namely, *Earphones, Phones* (mobile phones) and *Television sets*. These ontologies are diverse and contain a total of less than 10K concepts, and less than 20 unique kinds of relationships.

We picked ontologies of sub-domains that are based on related subjects shown in Table 3.1, and used them interchangeably as *source* and *target* ontologies to test BOLT-K. For instance, *Dengue* and *Malaria* are both vector-borne diseases transmitted by mosquitoes and share some causes, symptoms and effects, so we use them as a source-target pair. Likewise, *Alzheimers, Multiple Sclerosis, Depression* and *Anxiety* are mental health disorders; *Gymnosperms* and *Angiosperms* are two classes of plant varieties; *Earphones* are an accessory of *Phones*; and *Phones* and *Televisions* are electronic devices sharing some common properties. Hence, we also used these as source-target ontology pairs to test BOLT-K.

As mentioned in Section 3.3, we require a corpus of text documents containing sentences associated with the concepts in the source and target ontologies S and T. To fulfill this requirement for the bio-medical sub-domains, we used a combination of PubMed [40], PubMed Central (PMC) and the English Wikipedia. For an ontology subject $\mathfrak{C}$, we consider all those articles from PubMed, PMC and Wikipedia as part of a text corpus associated with $\mathfrak{C}$ if they contain the term $\mathfrak{C}$ either in their title or abstract. For the last three rows of product hierarchies in Table 3.1, we constructed a text corpus from the product information and descriptions extracted from the Amazon Product Dataset [120]. This dataset includes information about the numerous commodities sold online on `www.amazon.com`.

Table 3.1: Statistics of various domain ontologies used

Ontology sub- domain name	**No. of concepts**	**No. of concept pairs**	**No. of relations**	**Median no. of sentences per relation**
Dengue	5035	5923	11	6010
Malaria	2643	3556	11	5070
Alzheimers	5738	5961	2	9602
Multiple sclerosis	9036	11310	2	16518
Depressive disorder	2008	4576	3	3025
Anxiety disorder	1978	4194	3	3637
Gymnosperms	539	502	9	716
Angiosperms	306	302	10	590
Earphones	115	146	11	28
Phones	189	337	16	256
Television sets	72	87	11	8

3.5.2 Results

3.5.2.1 Identifying Related Concept Pairs

We first present in Figure 3.3 the performance of BOLT-K as well as two other baselines on the first step of identifying potentially related concepts for the target ontology, as described in Section 3.4.1. We consider each ontology in Table 3.1 as a target ontology.

Our first baseline (yellow bars of *context similarity* in Figure 3.3) associates a context with each target concept. This is a set of nearby words around the mention of the concept in the target text corpus. If a target concept has multiple mentions (and hence multiple contexts) in the corpus, we pick the context associated with a randomly selected mention. We then compute an embedding for each concept using a normalized term-frequency (*tf*-) weighted sum of the embeddings of its context terms. We use a pre-trained word2vec model [217] to get the term embeddings, ignoring the context terms that do not have embeddings in this model. The *tf*- weights are obtained from the frequencies of occurrence of the chosen context terms, in the available contexts associated with every mention of the target concept. Once

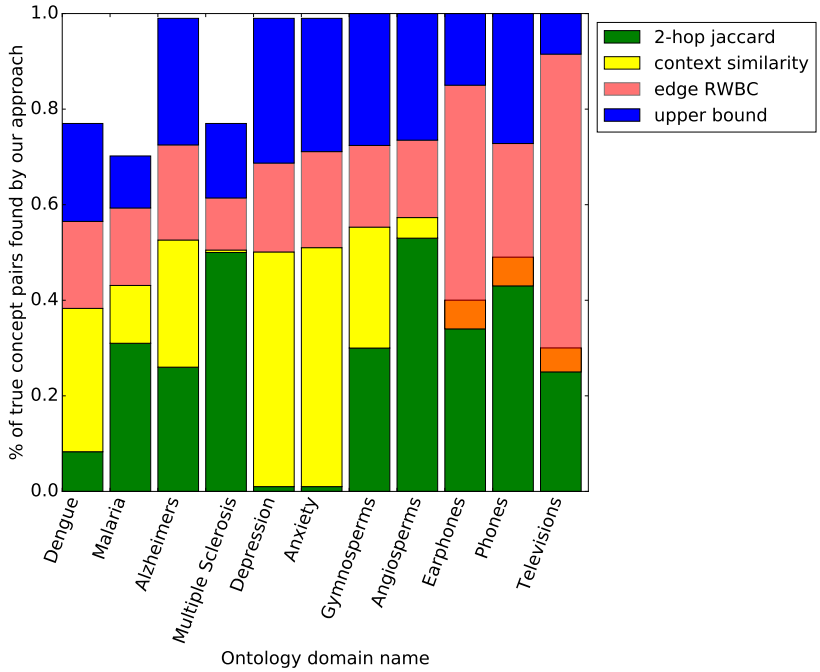


Figure 3.3: Performance of multiple approaches on identifying the concepts that are to constitute the target ontology

we generate an aggregated embedding of each concept, we compute the pairwise cosine similarity between all pairs of concept embeddings. We filter out all those concept pairs that do not have a similarity value above a learned threshold, and consider the remaining pairs as potentially related. Our second baseline (green bars of *2-hop jaccard* in Figure 3.3) estimates structurally similar concepts as possibly related to each other. It computes the 2-hop Jaccard index between pairs of connected concepts in the concept graph defined in Setion 3.4.1. For concepts c_1 and c_2, the Jaccard index is given by the number of neighbor concepts common to both c_1 and c_2 and reachable from them in 2 hops, divided by the union of the neighbors of c_1 and c_2. The pink bars of *RWBC* in Figure 3.3 denote the edge-based random walk betweenness centrality technique used in BOLT-K. The orange portions seen

in the last three bars show an overlap between the result numbers of the context-based similarity and the 2-hop Jaccard strategies.

The RWBC measure considers a more global view of the concept graph along with semantic attributes. We find that this causes a marked improvement over merely using either semantic information (the context similarity baseline), or the local structural neighborhood of concepts in the weighted concept network (the Jaccard index baseline). Our random walk betweenness centrality measure correctly identifies as *related*, about 60-75% of the linked concept pairs for the bio-medical ontologies. For the product hierarchies, it identifies more than 70% of the related concept pairs. We note here that every pair of related concepts that is present in the considered ontologies may not occur in the text corpora, i.e. may not be accompanied by textual information. There is thus a limitation on the number of potentially related concept pairs that can be found by our technique. The blue bars show this *upper bound* on the identifiable pairs of linked concepts for each ontology instance. This value is 76-80% for *Dengue, Malaria* and *Multiple Sclerosis*, and nearly 100% for the remaining hierarchies. Changing the scale of the result of BOLT-K's edge-based betweenness centrality strategy based on this upper bound, we observe an accuracy of more than 75% in finding potentially related concept pairs for the ontology instances.

3.5.2.2 Constructing Domain Ontologies via Knowledge Transfer

To the best of our knowledge, no existing work learns ontological relationships for a new target subject by levering prior knowledge from a distinct but related source subject. Thus, we compare our method with existing state-of-the-art techniques that only use information from the target domain. We evaluate all approaches on the task of ontological relationship prediction for the target ontology in Table 3.2. We reiterate that BOLT-K *does not* assume that the source and target ontologies have the same types of relationships. It uses at most 5

Table 3.2: Baseline F1 scores on the ontology pairs of Dengue ↔ Malaria, Alzheimers ↔ Multiple Sclerosis, Gymnosperms ↔ Angiosperms, Earphones ↔ Phones, and Phones ↔ Televisions. We use at most 5 target training sentences per relation type with data augmentation. t_{sr} is the target sub-domain sampling probability. The first sub-column of every S ↔ T source-target ontology pair denotes knowledge transfer from S → T and the second denotes S ← T (i.e. knowledge transfer from T → S).

Approach	**Deng ↔ Mal**		**Alz ↔ M.Scl**		**Depr ↔ Anxi**		**Gymn ↔ Angi**		**Earph ↔ Phone**		**Phone ↔ TV**	
	S→T	S←T	S→T	S←T	S→T	S←T	S→T	S←T	S→T	S←T	S→T	S←T
PCNN-Att	0.468	0.51	0.501	0.45	0.577	0.63	0.555	0.6	0.44	0.47	0.4	0.43
Path-Max	0.6	0.589	0.578	0.55	0.655	0.72	0.624	0.602	**0.501**	0.5	0.471	0.469
BLSTM-Att	0.601	0.62	0.505	0.49	0.567	0.6	0.592	0.63	0.43	0.45	0.41	0.399
ComplEx	0.561	0.626	0.54	0.5	0.66	0.701	0.58	0.54	0.48	**0.575**	0.514	0.455
BOLT-K (t_{sr}=0.6) - no Att	0.679	0.68	0.572	0.54	0.701	0.7	0.698	0.687	0.466	0.5	0.531	0.476
BOLT-K(t_{sr}=1)	0.66	0.64	0.502	0.53	0.582	0.68	0.631	0.67	0.47	0.5	0.5	0.47
BOLT-K(t_{sr}=0.8)	0.67	0.66	0.53	0.53	0.609	0.71	0.667	0.65	0.467	0.47	0.51	0.46
BOLT-K(t_{sr}=0.4)	0.644	0.63	0.545	0.5	0.65	0.66	0.63	0.62	0.44	0.45	0.46	0.44
BOLT-K(t_{sr}=0)	0.527	0.51	0.499	0.48	0.576	0.6	0.55	0.58	0.398	0.401	0.41	0.399
BOLT-K(t_{sr}=0.6)	**0.713**	**0.728**	**0.61**	**0.562**	**0.748**	**0.747**	**0.724**	**0.725**	0.487	0.53	**0.551**	**0.498**

relationship-labeled concept pair training examples per target domain. We compare BOLT-K with the following baselines:

1. PCNN-Att [392]: It employs a piecewise CNN with a sentence-level attention mechanism and distant supervision.

2. Path-Max [393]: It considers CNNs coupled with probabilistic relation paths learnt from the sentences between entities.

3. BLSTM-Att [401]: It uses a Bi-LSTM model without pooling, and with an attention mechanism different from ours.

4. ComplEx [328]: It uses low-rank matrix factorization to learn complex-valued embeddings for entities and relations. We found ComplEx to outperform multiple

other knowledge graph completion techniques like TransR [182], TransD [138], DistMult [376] and HolE [229], so we report only its results.

5. BOLT-K ($t_{sr} = 0.6$) - no Att: This is a variant of our model at a target sub-domain data sampling rate $t_{sr} = 0.6$ (the probability of the model being trained on data from the target sub-domain), on using max pooling without attention.

6. BOLT-K ($t_{sr} = t_{sr}$): The last five rows of Table 3.2 are variants of BOLT-K at target data sampling rates $t_{sr} = \{0, 0.4, 0.6, 0.8, 1\}$.

The first column of Table 3.2 shows the different approaches, and the subsequent columns show the ontology datasets. The $S \rightarrow T$ and $S \leftarrow T$ sub-columns for each column $S \leftrightarrow T$ indicate the direction of knowledge transfer, from ontology S to T and T to S respectively for our approach. The 'arrows' have no significance for the other approaches, merely indicating the ontology for which relations are being learnt (T in case of $S \rightarrow T$ and S in case of $S \leftarrow T$). These results have been computed after applying all augmentation strategies from Section 3.4.2 to the target training dataset.

We observe that BOLT-K obtains a 5-25% F1-score gain over the baselines in learning relationships for the different ontologies, with the best performance at a target training data sampling rate of 0.6. This reinforces our hypothesis of the utility of levering prior source knowledge in learning ontologies for a newly emergent target domain or sub-domain. BOLT-K is outperformed by ComplEx [328] by about 1% and 4% F1 score points respectively on the *Earphone* $\rightarrow$ *Phone* and the *Phone* $\rightarrow$ *Earphone* product ontology pairs. We believe this is because though these two products are somewhat related, their distinct characteristics may falsely inform our model and detract from its performance. BOLT-K performs better on the product pair of *Phones* and *Televisions* which share relatively more common attributes.

It is interesting to observe the difference in performance on interchanging the source and target subject ontologies. For example, there is more value in transferring knowledge from the *Phone* to the *Television* domain, compared to the reverse.

We further note that the overall performance of all approaches on the bio-medical ontologies is significantly better than on the commercial product ontologies. A reason for this could be the quality of textual data available. The text descriptions are often quite generic and repetitive across different product relation types (e.g. *"details about apple iphone 6 16gb - at&t - gold - great condition"*). The variation and uniqueness in context and sentence structure is much lesser than the text data for the bio-medical ontology datasets that come from research, news or encyclopedia articles. Since ComplEx utilizes less information from textual descriptions compared to the other methods, it performs better on the product datasets which have lower quality text input. We also analyze the topological structure of the concept graph (from Section 3.4.1) for the various ontology datasets. We find multiple well-separated connected components in case of the bio-medical ontologies, signifying a higher separability of their relationship categories. However, the product datasets have few (≤ 3) connected components for more than 10 relation categories. This implies that it is harder to distinguish between the product relationship types based on the current information.

3.5.2.3 Role of Target Training Data Size

Augmenting the limited amount of available target training data is a crucial step in our approach. Figure 3.4 (top) shows the change in F1-score of BOLT-K at the target data sampling rates $t_{sr} = \{0.6, 1\}$ for the *Dengue* $\rightarrow$ *Malaria* ontology pair. We compare BOLT-K to the best performing baselines from Table 3.2 (*PathMax* [393] and *ComplEx* [328]), as the number of training instances per relation type are increased. We also show in Figure 3.4 (bottom) an attempt to learn an ontology for *Angiosperms* using information from

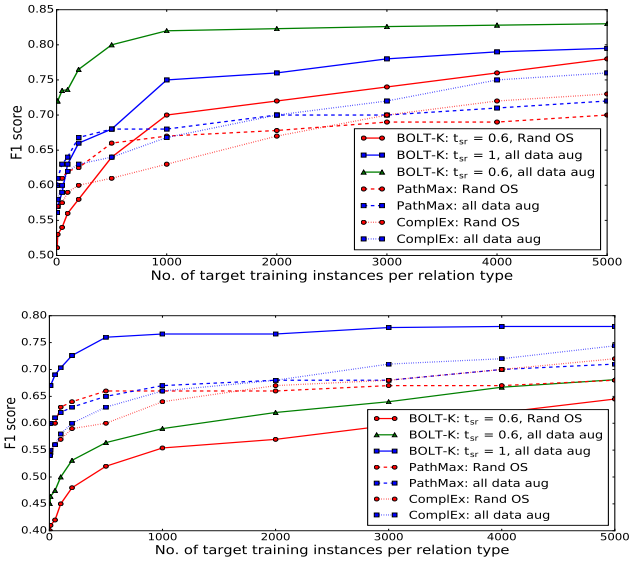


Figure 3.4: F1-score vs training data size for Dengue → Malaria (top) and Dengue → Angiosperms (bottom) ontology pairs.

a dissimilar and unrelated subject, *Dengue*. The solid, thick dashed and thin dashed line plots denote BOLT-K, PathMax and ComplEx respectively. The red plots use only random oversampling to augment the training data, while the blue and green plots use all strategies in Section 3.4.2. We observe that in case of *Dengue* → *Malaria* (top plot), as expected, the F1-score of each model rises with the amount of training data it receives. This gain is particularly significant at the left of the x-axis, from 5 to about 1000 training instances per relation category. Employing better data augmentation strategies than random oversampling contributes to the performance. Transferring knowledge from the source ontology lends us a consistent advantage of at least 5%, even at the maximum number of target training instances per relation type. In case of *Dengue* → *Angiosperms* (bottom plot), BOLT-K performs the best at $t_{sr} = 1$, i.e. without any input from the unrelated source (blue solid line plot). It obtains a 4% gain over ComplEx at the maximum number of training examples per relation

Table 3.3: Assessing various training data augmentation methods for BOLT-K, on the Dengue → Malaria ontology pair

Augmentation Technique	**Target F1 score**
Random oversampling (Rand OS)	0.511
SMOTE oversampling	0.458
. WordNet synonym based replacement + Rand OS	0.66
Word embedding based replacement + Rand OS	0.637
Trigram model based replacement + Rand OS	0.69
WordNet + Word embedding + Trigram + Rand OS	**0.713**

class. However, training input from the dissimilar source *Dengue* causes a performance drop (red and green solid line plots).

We next examine in Table 3.3 the impact of different data augmentation strategies on the performance of BOLT-K, on the *Dengue* → *Malaria* ontology pair. We observe similar trends on the other datasets as well. The first two rows show the performance using simple random oversampling and Synthetic Minority Oversampling (SMOTE) [47] on the target training examples, followed by the three augmentation strategies described in Section 3.4.2. The last row of Table 3.3 shows that these three strategies complement one another. Employing them all together with random oversampling gives us a performance advantage of about 4-8% over any one of them.

3.5.2.4 Role of Attention

Table 3.2 indicates that attentive pooling lends BOLT-K an F1 score gain of nearly 5%. Further drilling down, Figure 3.5 shows a heatmap of the attention weight values learned by BOLT-K for sentences belonging to different relation types. The cells of the heatmap contain the sentence word they represent. The background color highlights the importance of a word with respect to its neighboring context, and consequently, how it affects our model's decision.

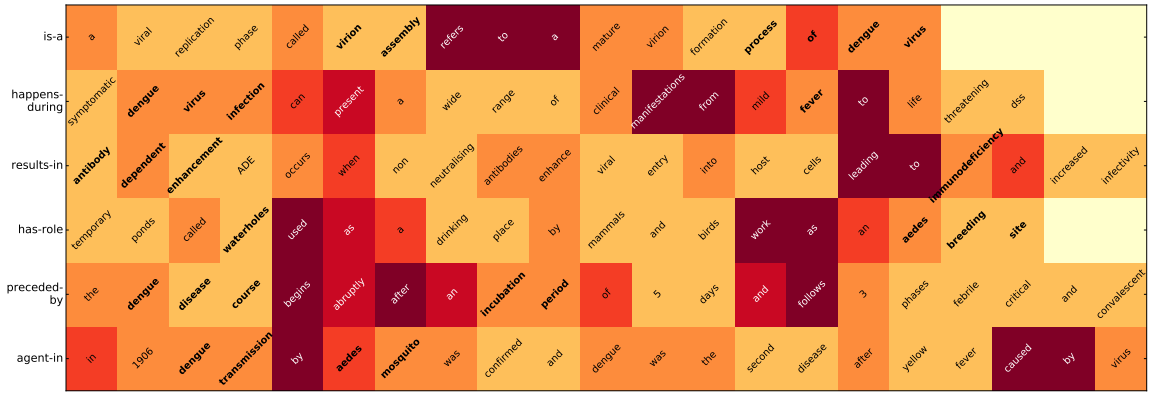


Figure 3.5: Visualizing word attention weights for sentences of certain relation types, for the Dengue ontology. The related concept phrase pairs in the sentence are in boldface, and a darker background corresponds to a higher attention weight.

The words in boldface are the concept phrases between which a relation is being predicted. For instance, the first row of the heatmap demonstrates that the words 'refers', 'to', and 'a' (dark background) are crucial in deciding the presence of an *is-a* relation between the concept phrases *virion assembly* and *process of dengue virus*. Similarly, in the fifth row, the words 'begins', 'after' and 'follows' influence the detection of the *preceded-by* relation between the concept phrases *incubation period* and *dengue disease course*. These examples show that BOLT-K has correctly learnt the relevant semantics needed to detect ontological relations between emergent concept phrases.

3.5.3 Discussion

We now dive deeper into the kind of ontological relationship information that our model can transfer across domains. Figure 3.6 displays the 2-dimensional t-SNE [201] visualizations of the representations learnt by BOLT-K for the sentence inputs of various relation classes. We observe that the relationships are largely well separable in case of the

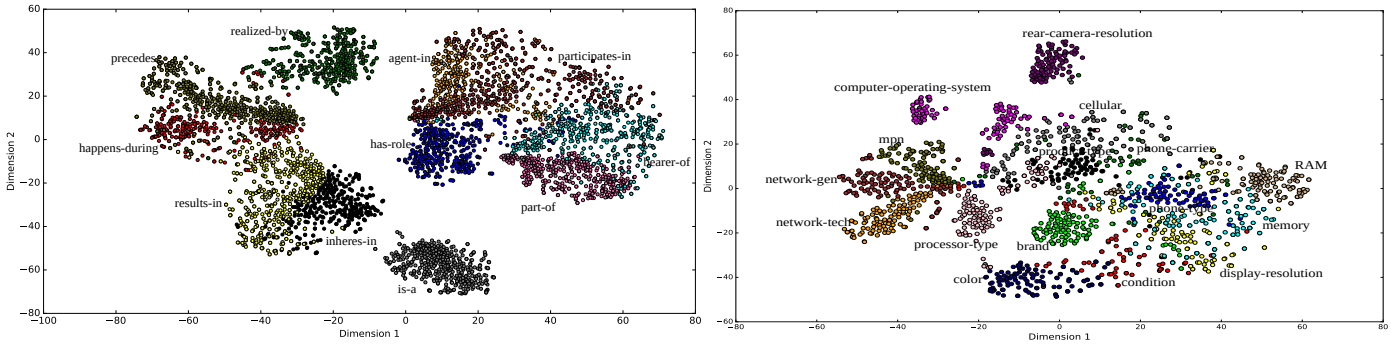


Figure 3.6: t-SNE visualization of the learnt sentence representations associated with the concept pairs for Dengue → Malaria (left) and Televisions → Phones (right).

Table 3.4: Frequently mispredicted relation types for a set of ontology pairs. For a relation type *r*, the parentheses show the percentage of mispredicted test instances of type *r*.

Source → Target	**Mispred. rel type: Frequently mispred. as**
Dengue → Malaria	*part-of* (35%): *is-a*; *inheres-in* (21%): *part-of*; *participates-in* (16%): *has-role*; *agent-in* (7%): *participates-in*; *bearer-of* (6%): *agent-in*
Malaria → Dengue	*part-of* (17%): *is-a*; *has-role* (17%): *part-of*; *happens-during* (8%): *part-of*; *precedes* (6%): *happens-during*
Alzheimers → Multiple Sclerosis	*type-of* (40%): *has-type*; *has-type* (38%): *type-of*
Multiple Sclerosis → Alzheimers	*type-of* (47%): *has-type*; *has-type* (41%): *type-of*
Depression → Anxiety	*is-a* (36%): *type-of*; *is-a* (12%): *has-type*
Anxiety → Depression	*is-a* (24%): *type-of*; *type-of* (18%): *has-type*
Gymnosperms → Angiosperms	*part-of* (19%): *is-a*; *adjacent-to* (20%): *part-of*; *has-part* (21%): *develops-from*; *located-in* (36%): *part-of*
Angiosperms → Gymnosperms	*preceded-by* (16%): *develops-from*; *develops-from* (11%): *part-of*; *has-participant* (11%): *part-of*; *part-of* (7%): *is-a*
Earphones → Phones	*brand* (42%): *phone-type*; *phone-type* (38%): *brand*; *display-res* (29%): color; *phone-carrier* (25%): *network-gen*; *RAM* (25%): *memory*
Phones → Earphones	*model* (44%): *product-type*; *brand* (30%): *model*; *color* (30%): *brand*; *additional-features* (22%): *compatibility*; *tagline* (20%): *brand*
Phones → Televisions	*refresh-rate* (39%): *display-type*; *model* (29%): *product-type*; *viewable-size* (29%): *total-size*; *display-res* (23%): *display-type*
Televisions → Phones	*RAM* (58%): *memory*; *cellular* (34%): *mpn*; *phone-type* (32%): *brand*; *display-res* (27%): *phone-type*; *cellular* (27%): *phone-carrier*

Table 3.5: Novel relation detection for Dengue → Malaria pair.

Relation type *r* left out during training	**% of left out concept pairs detected as novel**	**Novel concept pairs missed. *s*(x%) means x% concept pairs related by *r* were mispredicted as having relation *s*.**
is-a	76%	*part-of(8%), has-role(5%)*
part-of	51%	*has-role(19%), is-a(10%)*
happens-during	66%	*results-in(11%), part-of(6%)*
precedes	70%	*results-in(12%), happens-during(9%)*
results-in	61%	*precedes(14%), happens-during(9%)*
has-role	39%	*participates-in(21%), agent-in(17%)*
inheres-in	58%	*part-of(17%), bearer-of(10%)*
agent-in	41%	*participates-in(22%), bearer-of(19%)*
participates-in	46%	*has-role(23%), part-of(17%)*
bearer-of	64%	*agent-in(13%), part-of(9%)*
realized-by	67%	*results-in(9%), precedes(6%)*

Dengue → *Malaria* ontology pair (Figure 3.6 left), with semantically similar relations co-located in their vector space. For example, the relations *participates-in, agent-in, has-role, part-of* and *bearer-of* are situated nearby, and away from the relatively dissimilar relations *results-in* and *inheres-in*. We do not see as clear a segregation between the relation categories for *Televisions* → *Phones* (Figure 3.6 right). But we do find some relations alike in meaning close by in the vector space too, e.g. *RAM* and *memory*, *network-gen* and *network-tech*.

Table 3.4 shows the relation categories dominantly mispredicted by our model and the relation type that they are frequently mispredicted as, for different ontology datasets. It also reports for each mispredicted relation category *r*, the percentage of test instances of category *r* that were mispredicted. We find that a common reason for mispredictions is the high similarity in the meaning of certain relationships, due to which they can be used interchangeably in natural language. For instance, BOLT-K is unable to distinguish between the relations *is-a* and *type-of* in case of *Depression* ↔ *Anxiety*, *participates-in* and *has-role* in case of *Dengue* ↔ *Malaria* and *preceded-by* and *develops-from* in case of *Gymnosperms*

$\leftrightarrow$ *Angiosperms*. We also observe this trend in some cases of the product ontology pairs (last four rows of Table 3.4), such as *RAM* and *memory*, *brand* and *phone-type*, and *total-size* and *viewable-size*.

As part of further analysis, we seek to understand how well our model can recognize hitherto unseen relation types. For this purpose, we remove or "leave out" one relationship class (i.e. all concept pairs and sentences associated with it) completely from the training dataset of the target sub-domain (and from the training data of the source sub-domain if present). We train our model with the remaining training data as described in Section 3.4. We then use the Isolation Forest algorithm [188] with the authors' best case parameter settings to see if BOLT-K can detect concept pairs with the unseen relation type as *novel*, i.e. linked by a novel or unseen relationship. Isolation Forest takes as input the d-dimensional features obtained from BOLT-K's penultimate layer before the softmax layer. This novelty detection experiment shows our model's capability in identifying new or unseen relation categories which cannot be easily substituted by another, previously seen relation type.

The results are shown in Table 3.5, for the *Dengue* $\rightarrow$ *Malaria* ontology pair. The second column shows the fraction of concept pairs (whose relation class was left out while training) that have been detected as novel. For distinctive relationship types such as *is-a*, we find that more than 75% of the concept pairs which are linked by this relation in the ground truth are predicted as novel. But for left out relations which are not as semantically distinct from the other relationship types, BOLT-K is unable to satisfactorily identify that they were part of an unseen class. For instance, BOLT-K is unable to flag as novel more than 50% of the concept pairs linked by *has-role* and *participates-in*, when they were left out during training. For the concept pairs that were connected by the left-out relationship yet were not detected as novel, we investigated the relationship categories that BOLT-K was predicting them as.

The third column of Table 3.5 shows the two dominant relation types that most concept pairs (linked by the left-out relationship) are mispredicted as belonging to. We found these predictions to be largely logical, when compared with the actual ground truth relationships. For example, 17% of the concept pairs linked by an *inheres-in* relation that was unseen during training were mispredicted as being connected by a *part-of* relation. 21% of the concept pairs connected by a *has-role* relation were mispredicted as having a *participates-in* relation. These mispredictions are semantically plausible due to the similarity in meaning of the misunderstood relationship groups.

3.6 Conclusion

We present BOLT-K, a Bi-LSTM framework with attentive pooling. It identifies concepts relevant to newly emerging subjects or events in various domains from their textual accounts, and automatically learns structured ontologies for them. We bootstrap this process by transferring knowledge from an existing, related sub-domain. We also leverage training data augmentation to accentuate the limited expert-labeled data available for these emergent sub-domains, at no further annotation cost. We extensively evaluate BOLT-K on real-life bio-medical and commercial product ontologies.

Chapter 4: FACE-KEG: Fact Checking Explained Using Knowledge Graphs

Accurately structuring and representing online information in the form of ontologies, taxonomies or knowledge graphs, as we proposed in Chapters 2 and 3, requires the input information to be factually accurate. In recent years, a plethora of fact checking and fact verification techniques have been developed to detect the veracity or factuality of online information text for various applications. However, limited efforts have been undertaken to understand the interpretability of such detection, i.e. explaining *why* a particular piece of text is factually correct or incorrect. In this Chapter, we seek to bridge this gap by proposing a technique FACE-KEG, to perform *explainable* fact checking via an encoder-decoder setup. Given an input fact or claim, our proposed model constructs a relevant knowledge graph from a large-scale structured knowledge base, which it encodes via a novel graph transforming encoder. Our model also retrieves and encodes relevant textual context about the input text from the knowledge base. FACE-KEG then jointly exploits both the concept-relationship structure of the knowledge graph as well as semantic contextual cues in order to (i) detect the veracity of an input fact, and (ii) generate a human-comprehensible natural language explanation justifying the fact's veracity. We conduct extensive experiments on three large-scale datasets, and demonstrate the effectiveness of FACE-KEG while performing fact checking. Automatic and human evaluations further show that FACE-KEG significantly

outperforms competitive baselines in learning concise, coherent and informative explanations for the input facts.

4.1 Introduction

Huge volumes of data are continuously being generated and extracted as a result of the rapid development in information extraction techniques. Detecting if a given piece of information is *true* or *factually correct* plays a vital role in various natural language applications such as language understanding, knowledge graph completion and open domain question answering [48, 89, 285]. A plethora of techniques have been developed to tackle the problem of automated *fact checking* [14, 80, 89, 230, 247, 285, 287, 289, 324]. However, most approaches merely focus on *detecting* if a claim is true. They cannot adequately explain *why* a claim was detected as true or false. Such explanations are desirable because they can (i) help non-experts comprehend the veracity of niche or domain-specific claims; and (ii) provide new and useful insights that can improve the general performance of fact checking. As an example, rather than just detecting that the claim *"Due Date is a horror film"* (from [325]) is false, it is useful to have an accompanying explanation such as *"Due Date is a 2010 American comedy film..."*. Fact checking websites such as *www.politifact.com* and *www.snopes.com* exist that explaining the veracity of input claims by providing supporting or refuting evidence. However, they involve expensive and time-consuming checks and assessments by human reviewers. In this work, we seek to *automate* the process of explaining fact checking decisions.

A step towards explainable fact checking is solving the task of *fact verification* [51, 114, 230, 285, 324, 325, 400]. It attempts to verify whether an input claim is supported or refuted by a given piece of evidence text. Unlike our work that seeks to *generate* concise

explanatory evidence, the fact verification task takes the evidence text as input. Reasoning about fact checking decisions via extractive approaches that retrieve evidence sentences about a claim has achieved promising results [14, 89, 230, 289, 400]. Though such evidence can serve as an explanation, it is often long-winded and contains irrelevant information unnecessary to justify the claim. In many cases, the retrieved evidence contains text that is semantically related to the input claim, but cannot explain its factuality. Efforts have also been undertaken to explain fact veracity via (i) semantic traces and patterns derived from large, general-purpose knowledge graphs (e.g. DBPedia [12]); (ii) formal logic rules; and (iii) attributed relations from semi-structured tables or databases [51, 80, 89, 287]. However, explanations formulated in this fashion are often incoherent, longer than necessary with extraneous information, and are not as easily readable and interpretable as plain text. We desire to address these limitations in our work by detecting the veracity of input claims or facts, and simultaneously learning to explain their veracity in a concise, coherent, easily comprehensible manner.

To this end, we propose a framework FACE-KEG. It first builds a knowledge graph, and retrieves unstructured textual context pertaining to each input claim. A novel graph transformer network and a bidirectional RNN network are employed to encode the knowledge graph and textual context respectively. This is followed by jointly training a classifier that predicts if the claim is true or not, and a decoder that learns to generate a natural language explanation clarifying the veracity of the claim. We propose to learn explanations for input claims from the interrelated perspectives of the claim content, suitable background context and structured conceptual knowledge relevant to the claim. Apart from gaining a semantic understanding, analyzing pertinent context can help unearth indirect cues such as the stance or sentiment of the input fact or claim, that can be useful for fact checking [14, 253]. We

also seek to introduce open-domain, open-topic auxiliary knowledge in our FACE-KEG framework via general-purpose knowledge bases, that (i) is not restricted to small-scale domain-specific information [36, 162]; and (ii) leverages the rich graphical structure of linked knowledge entities, instead of linearizing them [5, 80, 89, 163, 287]. To the best of our knowledge, this is the *first* attempt in the literature to explain fact checking by directly generating textual explanations clarifying the veracity of input facts in an *abstractive* manner. To summarize, the main contributions of our work are:

- FACE-KEG provides a fresh perspective to explainable fact checking by jointly encoding structured and unstructured contextual knowledge, to generate *abstractive* explanations directly comprehensible by humans.
- FACE-KEG advances over explanations from past work based on (i) verbose evidence text with many extraneous details (e.g. FEVER task, fact checking website pages); (ii) knowledge entity pattern sequences; or (iii) logic rule sequences.
- FACE-KEG significantly outperforms state-of-the-art baselines, as per both automated and human evaluation measures.

4.2 Related Work

Fact Checking, Verification and Explainability: *Automated fact checking* [324] has been studied from several perspectives over the last few years, e.g. as natural language inference (NLI) [230,238,355], multimodally using images [305], reasoning from relational tables [51], textual entailment [355], and as the closely related problem of fake news or misinformation detection [285]. Unlike our proposed work, FACE-KEG, most techniques only detect if an input claim is true or false [14, 114, 230, 247, 285, 324, 400] without justifying their

decision. *Fact verification* aims to verify if certain input 'evidence' can justify the veracity of a claim [114, 325, 400]. Many techniques [44, 114, 190, 204, 230, 386, 400] have been proposed to solve the popular FEVER shared task for fact verification [325, 326]. However unlike these methods, FACE-KEG does not take as input any explanatory or evidence text associated with the input claims.

Logic rules as well as patterns derived from external knowledge bases like Wikipedia have been used to detect and explain fact veracity [80, 89, 287]. Extractive explainable fact checking approaches [10, 289, 378]. also take additional 'evidence' text as input (e.g. online user comments, fact checking web pages, explanatory comments from experts), and output a relevant subset of this text as an explanation. On the other hand, FACE-KEG actually generates concise, abstractive explanations without taking any of the actual explanatory text as input. This is in contrast to lengthier explanations with extraneous details (e.g. in FEVER task or fact checking website pages); or non-human readable explanations (e.g. knowledge entity patterns, logic rule sequences). Since it is not always possible or feasible to obtain such additional, human-curated evidence information per claim, we do not compare FACE-KEG against the above extractive approaches. Thus, through this work we make a preliminary attempt to automate the generation of explanatory justifications provided by humans for fact checking.

Knowledge Enhanced Text Generation: Past work has used contextual cues from external knowledge bases in both structured and unstructured forms to enhance natural language generation. Contrary to FACE-KEG which employs a large scale knowledge base with generic entities, concepts and relationships; some studies [162, 399] utilize small, domain-specific knowledge bases with a limited number of relation types (e.g baseline *KBLLH'19* in Section 4.4). This renders them difficult to adapt for open-domain, open-topic text

generation. Prior work has modeled external knowledge entities both individually and as linear sequences of paths; failing to take their rich connectivity structure into account [5, 80, 89, 163, 287] (e.g baseline *FACE-KEG-linear enc.* in Section 4.4). We find that modeling the external structured knowledge as a graph improves text generation. Graph convolutional networks [109, 206], graph attention networks (GATs) [349, 399], graph neural networks (GNNs) [109], gated GNNs [17, 296] and more recently, graph transformer networks [36, 162, 391] have been proposed to directly encode graph inputs. In this work, we propose a graph transformer network comparable to the graph-to-sequence model proposed by Cai et al [36] (baseline *CL'20* in Section 4.4). However, our work differs in the following ways: (i) Cai et al assume the presence of largely connected, dense knowledge graphs with root vertices, unlike the sparse, disconnected knowledge graphs that FACE-KEG works with. (ii) Unlike FACE-KEG, they do not employ additional textual context that can potentially enhance our understanding of the input claim, while performing text generation. (iii) Cai et al require the edge label (relation) information from their graphs to be explicitly available and encoded in their model, unlike FACE-KEG, since such information (if any) is much sparser in our knowledge graphs.

4.3 The FACE-KEG Framework

Formally, given a textual claim or *fact* and an external knowledge base; our goal is to investigate the *veracity* of the fact (i.e. identify if it is true or not) and generate a human-comprehensible explanation clarifying the truthfulness of the fact. Figure 4.1 displays the overall architecture of FACE-KEG. We construct a knowledge graph associated with the input claim (Section 4.3.2). We then extract textual context relevant to the claim from the available knowledge base, as described subsequently. A bidirectional RNN (Section 4.3.1)

and a graph transformer network (Section 4.3.2) are employed to encode the associated textual context and knowledge graph respectively. This is followed by jointly training a classifier that predicts if the input fact is true or false, and a decoder that learns to generate a natural language explanation about the veracity of the fact (Section 4.3.3). We employ both the structured knowledge graph and the unstructured textual context in FACE-KEG so that the two complement each other to perform explainable fact checking. This ensures that even if sufficient background information needed to assess the veracity of a fact is not present in the constructed knowledge graph, the extracted context will compensate for it, and vice versa.

We assume that we have available a large, structured knowledge base to help understand background information about the given facts. The implicit assumption here is that the chosen knowledge base contains sufficient supporting information to help predict and justify the veracity of the facts under consideration. We use the DBPedia ontology [12] (and its associated Wikipedia text corpus) as the knowledge base in this work. For each claim or fact, we first perform *entity linking* using a constituency parser [92] to extract all named entities $\mathscr{E}$ belonging to DBPedia from the fact text. We also extract relevant background context from the knowledge base regarding the claim, to enhance our understanding of the claim [114]. For each input claim, we search for Wikipedia documents relevant to the named entities extracted from the claim via the MediaWiki API[6]. This retrieved document set D for each claim is filtered based on the word overlap between the document titles and the claim text. We next select the sentences most relevant to the claim from all the sentences present in the retrieved documents, by learning a modified version of the Enhanced Sequential Inference Model (ESIM) [50]. Ten ESIM models with different parameters are trained via hinge loss

[6] `https://www.mediawiki.org/wiki/API:Main_page`

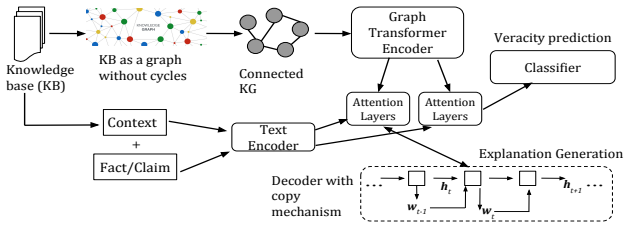


Figure 4.1: The pipeline of our FACE-KEG framework.

to generate a relevance score for each claim-sentence pair. The mean of the scores from the ten models is computed for each claim-sentence pair. Finally, the top five highest ranked sentences are considered as the unstructured context for that claim.

We obtain additional context via distant supervision [214]. For each claim, we examine if there exists a path between pairs of claim entities within the *knowledge graph* (Section 4.3.2) associated with that claim. For all such pairs of claim entities connected by a path, we search the above obtained document set D for sentences that contain mentions of both entities in the pair. We add all such sentences (if any) to the context associated with the input claim.

We next describe each component of FACE-KEG in detail.

4.3.1 Text Encoder

We employ a sequential text encoder to learn representations for the claim text, as well as for the unstructured textual context associated with the claim. Following the immense success of pre-trained language representation models for various NLP tasks, our encoder consists of an ELMo [250] embedding module to transform the words in the input text into a vector sequence. We add a bidirectional LSTM [106, 123] layer on top of the representations learnt by ELMo, as per [252]. The last hidden state of the bi-LSTM run over each word in the input text generates vector representations $\mathbf{R}_1$ for the claim text, and $\mathbf{R}_2$ for the

associated context. The actual network weights of ELMo are frozen while training, except for those used to weight the outputs from the different LSTM layers of ELMo. The final output text encoding $\mathbf{T}$ for each claim is the sum of $\mathbf{R}_1$ and $\mathbf{R}_2$.

4.3.2 Knowledge Graph Transformer Encoder

Knowledge Graph Construction: We first represent the entities and relationships within DBPedia ontology as an acyclic graph [307]. We connect the DBPedia entities $\mathscr{E}$ associated with the input fact (extracted earlier) into an undirected knowledge graph $G = (V, E)$. The nodes V in G represent the entities, while the edges E between them are based on their ontological relationships from DBPedia. Unlike most previous work, an important part of our graph construction technique is that it does not restrict the interactions of a node to only those nodes that it is directly connected to. For this purpose, we retrieve from the original DBPedia knowledge ontology all those entities $\mathscr{E}_1$ occurring in the respective fact's textual context, all entities $\mathscr{E}_2$ that are linked to each entity in $\mathscr{E}$, and all the entities $\mathscr{E}_3$ that are linked to each entity in $\mathscr{E}_2$. $k_1\%$ randomly sampled entities from $\mathscr{E}_1$, $k_2\%$ randomly sampled entities from $\mathscr{E}_2$ and $k_3\%$ randomly sampled entities from $\mathscr{E}_3$ are each added to the node set V. Entity nodes that are related to each other according to the DBPedia ontology are connected by an edge in G. Sampling keeps a control on the graph size, and ensures that too much extraneous information is not added into the knowledge graphs associated with the input claims. Our use of the first order as well as second order entity neighborhoods of the relevant named entities empowers FACE-KEG to capture essential local information, and also establish global connections between distant nodes through intermediate entity paths. In case the resulting graph G is disconnected, we find the connected components within it [322]. We then add an additional 'global' node to G, and connect it to a randomly

chosen node in each connected component of G, to sustain flow of information within the graph. We construct such a knowledge graph for each input claim whose veracity is to be determined and explained.

Encoding Graph Nodes: We propose a *graph transformer* encoder model to learn representations for the knowledge graphs constructed for each input fact (Figure 4.1). Unlike GNNs and GATs, transformer [334] style architectures enable the learning of dependencies between nodes irrespective of the distance between them in the input graph. We employ self-attention to compute hidden representations of each node in the graph $G = (V, E)$, by attending over their first-order neighborhoods in G. Apart from accounting for these local node interactions, we also stack together multiple blocks of graph attention modules that allow information to propagate globally throughout the graph. This information propagation as well as the transformer attention mechanism [334] enables FACE-KEG to learn global context patterns from the graph structure.

As mentioned earlier, the nodes in G correspond to DBPedia entity names and can consist of a single word or multiple words. We begin by employing the text encoder from Section 4.3.1 to learn a d-dimensional initial representation for each entity node in V (excluding the global node). This initial representation matrix learnt for all graph nodes in V serves as the input to our proposed graph transformer encoder in Figure 4.1. For each node representation $\mathbf{x}_i \in \mathbb{R}^d$, a multi-headed self-attention mechanism is applied over its first-order neighborhood $\mathcal{N}_i$ (nodes to which x_i is connected by an edge in G), to produce representation $\mathbf{x}_i^{ATT}$ as follows:

$$a^n(\mathbf{q}_i, \mathbf{k}_j) = \frac{exp((\mathbf{W}_K \mathbf{k}_j)^T \mathbf{W}_Q \mathbf{q}_i)}{\sum_{z \in \mathcal{N}} exp((\mathbf{W}_K \mathbf{k}_z)^T \mathbf{W}_Q \mathbf{q}_i)} \tag{4.1}$$

$$\alpha_{ij}^{n} = a^{n}(\mathbf{x}_i, \mathbf{x}_j) \;\; ; \quad \mathbf{x}_i^{ATT} = \mathbf{x}_i + \bigoplus_{n=1}^{N} \sum_{j \in \mathscr{N}_i} \alpha_{ij}^{n} \mathbf{W}_V^{n} \mathbf{x}_j \tag{4.2}$$

Here a^n denotes the n-th self-attention head, whose attention function learns independent trainable projection matrices $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{dxd}$ with respect to input parameters $\mathbf{q}$ and $\mathbf{k}$ respectively [334]. The resulting product is normalized across all connected edges in the neighborhood $\mathscr{N}_i$. We scale the dot products by their dimensionality $\frac{1}{\sqrt{d}}$ to prevent small gradient values due to large dot product magnitudes [334]. $\bigoplus$ denotes the concatenation of N independent attention heads before a residual connection is applied, α_{ij} represents attention scores, $\mathbf{W}_V^n \in \mathbb{R}^{dxd}$ is a weight matrix to be learned, and $\mathbf{x}_i$ and $\mathbf{x}_j$ denote initial representations of nodes in G.

This multi-headed attention module in the graph transformer is then followed by a fully connected two-layer feed-forward network FFN(y), with a non-linear transformation $f(\mathbf{W_1}y + b_1)\mathbf{W_2} + b_2$, between layers. Both the above two blocks use a residual connection followed by a normalization layer at the output for node i:

$$\mathbf{x}_i^{OUT} = \text{Norm}(\text{FFN}(\text{Norm}(\mathbf{x}_i^{ATT}) + \text{Norm}(\mathbf{x}_i^{ATT})) \tag{4.3}$$

The attention and FFN blocks are stacked L times in the transformer to promote information propagation through the graph. The output $\mathbf{x}_i^{OUT,l}$ of the previous transformer layer l is used as the input $\mathbf{x}_i^{l+1}$ of the next layer $l+1$ for node i. The encoding of all nodes in the knowledge graph G is thus represented by the representation matrix $[\mathbf{x}_i^L]$ output by the graph transformer encoder.

4.3.3 Explainable Veracity Prediction

Once suitable encoded representations have been generated for the input fact, its associated knowledge graph and relevant context, we utilize these to (i) determine whether the input fact is true or not (veracity prediction); and (ii) generate a textual explanation clarifying the veracity or falsity of the fact. This is a step ahead of most prior efforts in the literature, that only identify the veracity of facts without explaining the reason behind their decisions. We jointly train FACE-KEG for both these tasks. The respective encodings $\mathbf{T}$ and $\mathbf{X}^L$ learnt by the text encoder (Section 4.3.1) and the knowledge graph transformer encoder (Section 4.3.2) are attended over independently by the attention function $a()$ defined in Equation 4.1:

$$\alpha_{G,i} = a(\mathbf{h}_t, \mathbf{x^L}_i) \;\; ; \;\; \mathbf{o}_G = \mathbf{h}_t + \bigoplus_{n=1}^{N} \sum_{i \in V} \alpha_{G,i}^n \mathbf{W}_G^n \mathbf{x^L}_i \tag{4.4}$$

Here $\mathbf{h}_t$ denotes the last hidden state of the knowledge graph transformer encoder, $\alpha_{G,i}$ represents attention weights and $\mathbf{W}_G^n \in \mathbb{R}^{dxd}$ indicates a weight matrix to be learned. The representation vector $\mathbf{o}_T$ from the claim and context sequences is computed similarly by attending over the encoding $\mathbf{T}$ learnt by the text encoder. To perform veracity prediction, the two representations $\mathbf{o}_G$ and $\mathbf{o}_T$ are passed through attention layers [196] to get a final aggregated representation $\mathbf{o}$, which is then passed to a classifier. It consists of dense ReLU layers and a softmax layer to predict veracity (Figure 4.1).

We next outline how to simultaneously generate a natural language explanation justifying the veracity of the input fact. For this, we train a sequential LSTM decoder with attention [196], and a copy mechanism [282] for copying inputs from the knowledge graph and the text sequence. It shares the same graph transformer encoder and text encoder as the veracity prediction classifier. The representation of the global node is used to initialize

the hidden states of the decoder at each time step, analogous to the final encoder hidden state in a traditional sequence-to-sequence encoder-decoder framework. At each decoding time step t, we use the decoder's hidden state $\mathbf{h}_t$ and multi-headed attention to compute context vectors $\mathbf{o}_G$ and $\mathbf{o}_T$ for the graph and text sequence respectively. This computation and the learning of the attention scores and weight matrices is done in the same way as while performing veracity prediction (see Equation 4.4). The only difference is that $\mathbf{h}_t$ now represents the decoder's hidden state. We next construct the final context vector $\mathbf{o}_t$ of the decoder at time step t by concatenating the two vectors $\mathbf{o}_G$ and $\mathbf{o}_T$. Both the hidden state $\mathbf{h}_t$ and decoder context $\mathbf{o}_t$ are then passed to the next time step of the decoder. Using a copying mechanism while decoding addresses the data sparsity issue in word token prediction, and facilitates the generation of specific terminologies or named entities relevant to the input fact. Our model thus either generates tokens from a pre-defined vocabulary, or directly copies them from the model input with probability p [282]:

$$p = \sigma(\mathbf{W}_{copy}[\mathbf{h}_t \oplus \mathbf{o}_t] + b_{copy}) \tag{4.5}$$

where σ denotes the sigmoid function. Hence, the probability distribution $P(w_t)$ of generating the next target token w_t by generating a word from the vocabulary or copying a word is given by:

$$P(w_t) = p * dist_{copy} + (1-p) * dist_{vocab} \tag{4.6}$$

$$dist_{i,copy} = a([\mathbf{h}_t \oplus \mathbf{o}_t], \mathbf{y}_i) \tag{4.7}$$

where $dist_{copy}$ is the probability distribution over the named entities and input tokens for $\mathbf{y}_i \in \mathbf{X} \oplus \mathbf{T}$, and $a()$ is the attention function from Equation 4.1. The distribution $dist_{vocab}$ is

computed by scaling the concatenation $[\mathbf{h}_t \oplus \mathbf{o}_t]$ to the vocabulary size and taking a softmax over it. The output sequence of words w_t generated in the above manner by the decoder at each time step t form the natural language explanation for the input claim.

Training and Implementation Details: Both the veracity detection classifier and the explanation generation decoder share the same knowledge graph transformer encoder and text encoder. The loss function of the decoder $\mathscr{L}_D$ minimizes the negative joint log likelihood of the target text vocabulary and the copied entity indices. The classifier loss function $\mathscr{L}_C$ minimizes the binary cross entropy across both classes. The final objective for jointly training both components is given by $\mathscr{L}_D + \beta\mathscr{L}_C$, where β is a tunable hyperparameter controlling the trade-off between the two loss functions. All parameter values in FACE-KEG have been tuned based on performance on a validation set. The values of sampling parameters k, k_1 and k_2 while constructing the knowledge graph are set to 25%, 25% and 10% respectively. The number of hidden states of the LSTMs and the dimensionality of the attention layers are set to 500. The feed forward network block in the knowledge graph encoder has a PReLU [118] activation function and an intermediate size of 2000 units. The attention and FFN blocks are stacked $L = 6$ times in the graph transformer. We use dropout [298] with a probability value of 0.3 in the self attention layers, and the number of attention heads is set to 4. We use beam search [311] in the explanation generation decoder, with a beam size of 4. While training to optimize the loss function, we use SGD with momentum [256] and warm restarts, with an initial learning rate of 0.25. We train FACE-KEG for 20 epochs with early stopping based on the validation loss.

Table 4.1: We show the number of claim instances; average length of claims, contexts and explanation texts; and average knowledge graph size *per claim* (#vertices, #edges).

Dataset	**#claims**	**Avg claim,context,expl. len**	**Avg KG(V,E)**
FEVER	123K	8, 122, 28	(30, 42)
MultiFC	20K	14, 156, 73	(18, 31)
FakeCOVID	3K	18, 169, 61	(45, 66)

4.4 Evaluation

4.4.1 Experimental Setup

Datasets: We evaluate the performance of FACE-KEG on three large, challenging fact checking datasets described in Table 4.1: (i) FEVER shared task data [325,326]; (ii) MultiFC dataset containing real-world fact checking claims from multiple domains [13]; and (iii) FakeCOVID dataset containing cross domain facts on the COVID-19 pandemic [284]. The total number of DBPedia entities with respect to the facts present in these three datasets is 966K, with 735K links among these entities. Claims for which sufficient evidence is not available to justify their veracity are eliminated from consideration. For each dataset, we split the claims such that 70% are used for model training, 15% for validation and 15% for testing. The FEVER dataset contains human-annotated ground truth explanations that support the veracity labels for each claim. However, the MultiFC and FakeCOVID datasets *do not* provide direct explanations for the claims. They provide links to fact checking web pages whose content can justify the claims' veracity labels. Therefore, to acquire ground truth explanations for these two datasets, we filter out the metadata, extract the web page content, and utilize a tree-based convolutional neural network with heuristic matching [13]. It chooses the top four ranked evidence sentences from the associated web pages as the true explanations for the claims. We use a random sample of claim-explanation pairs to

manually verify that sensible ground truth explanations are obtained for these two datasets. We assume that the true explanation accompanying each claim is authentic and reliable. We also eliminate all sentences in our extracted input textual context (from Section 4.3) that overlap with the ground truth explanation (if any). Finally, for explainable fact checking, FACE-KEG requires sufficient context relevant to the input claim, as well as good coverage of named entities (associated with the input claim) in the external knowledge base being used. DBPedia and Wikipedia work well for the FEVER and MultiFC data. For the FakeCOVID data, we employ an additional news corpus[7] as a source of useful context for many claims which are underrepresented in Wikipedia. It contains 1.2M news articles on COVID-19.

Baselines: To the best of our knowledge, our work is the first to perform explainable fact checking by generating *abstractive* natural language explanations for the veracity of input claims. We thus evaluate both the tasks of predicting fact veracity (Task I), and simultaneously generating meaningful explanations to support the claimed veracity (Task II). For Task I, we compare FACE-KEG with:

(1) The **FEVER** shared task baseline [325, 326];

(2) FEVER leader board methods: **Athene** [114], **UCL MR** [230], **UNC** [386], **Papelo** [204], **DOMLIN** [301], **GEAR** [400], **KGAT** [190].

For Task II, we compare FACE-KEG's explanation learning module with state-of-art text generation methods, using the same knowledge graphs, text corpora and context (in Section 4.3) for all methods.

(1) **BHC'18** [17]: encodes the input knowledge graph via a GNN.

(2) **KBLLH'19** [162]: their graph transformer encoder and knowledge graph construction process are different from FACE-KEG. Unlike FACE-KEG, their graphs need explicit

[7]https://blog.aylien.com/free-coronavirus-news-dataset/

relation labels between all entity pairs, which are then added as nodes into the graph. This increases the graph size fast for even a moderate number of entity-relation types, hurting performance. Also, their graphs only contain entity-relation edges, and no direct connections between entities.

(3) **CL'20** [36] and (4) **YWW'20** [384]: both use graph transformer encoders with a different architecture than FACE-KEG, and transformer decoders unlike the RNN decoder of FACE-KEG. They formulate their models assuming much denser, rooted, connected graphs with dense edge relation label information; unlike the sparse, disconnected knowledge graphs in our fact checking problem scenario.

(5) **FACE-KEG-only our KG enc.** and (6) **FACE-KEG-only our text enc.**: variants of FACE-KEG using only our proposed graph transformer encoder and only our text encoder, respectively;

(7) **FACE-KEG-no context**: variant of FACE-KEG where the text encoder only encodes the claim, and no supporting context;

(8) **FACE-KEG-single att.**: varying FACE-KEG by passing our proposed graph and text encoders through a single attention block, in lieu of two sets of attention layers (Figure 4.1) before prediction;

(9) **FACE-KEG-linear enc.**: variant of FACE-KEG that ignores graph structure. It uses a bi-LSTM to sample and encode linear entity paths to represent the graph [5], instead of our graph encoder;

(10) **FACE-KEG-GAT enc.**: variant of FACE-KEG using a GAT [349] with PReLU activations stacked between self-attention layers to encode the input knowledge graph, in lieu of our graph encoder.

(11) **FACE-KEG-GTN enc.**: replaces our proposed graph encoder with a different graph transformer network (GTN) [391], that learns multiple meta-path graphs from the input graph, and performs convolutions on them to get an aggregated graph encoding.

Performance Metrics: To evaluate Task I, we use the classification metrics of accuracy and F1-score. For Task II, we use BLEU [237], METEOR [15], ROUGE [180] and Entity Overlap (compares the overlap of named entity phrases between the generated and true explanations). We test the difference between the best approach for each experiment and all other approaches using the approximate randomization test [261] with the Bonferroni correction for multiple comparisons, at a confidence level of 0.05 ($p < 0.05$). We also assess the quality of generated explanations via a user study (Table 4.5).

4.4.2 Results and Analysis

Task I Evaluation: Table 4.2 evaluates predicting the veracity of the test input claims for the three datasets. We observe an improvement for FACE-KEG of at least 2% in terms of F1-score and 3% in label accuracy, compared to different systems from the FEVER shared task leader board (first eight rows of Table 4.2). The proposed FACE-KEG (last row) and the recently proposed GEAR [400] and KGAT [190] are competitive on the FEVER dataset, with FACE-KEG significantly outperforming them on the other two datasets. *This is a remarkable result since unlike the baselines, FACE-KEG does not take any evidence or explanation text as input and more importantly, independently explains its results in Task II below.* The ninth to sixteenth rows of Table 4.2 show that FACE-KEG as proposed outperforms its variant baselines described earlier by at least 4%, in terms of label accuracy and F1 score. The decrease in both these metrics on using only the graph encoder or only

Table 4.2: Claim veracity prediction on the three datasets; with best results shown in bold. † shows *no* statistically significant difference from the best results, and is also in bold.

Approach	**Label accuracy %** **(FEVER/M.FC/F.COV)**	**F1 score %** **(FEVER/M.FC/F.COV)**
FEVER task baseline [325]	53.9 / 48.1 / 56.2	51.5 / 44.3 / 55.1
Athene [114]	69.8 / 66.2†/ 68.3	66.1 / 63.4 / 66.6
UCL MR [230]	68.2 / 64.9 / 66.8	64.3 / 63.6 / 63.2
UNC [386]	71.3 / 65.7 / 69.1	69.4 / 63.9 / 66.5
Papelo [204]	64.6 / 61.04 / 59.9	62.9 / 58.8 / 57.4
DOMLIN [301]	67.8 / 64.6 / 62.9	65.3 / 61.7 / 60.2
GEAR [400]	**73.9**† / 64.1 / 71.8	71.1/ 63.3 / 70.4
KGAT [190]	**74.2** / 64.7 / 73.02	**71.4**/ 63.1 / 70.8
FACE-KEG-only our KG enc.	65.5 / 57.7 / 64.8	62.1 / 56.1 / 63.2
FACE-KEG-only our text enc.	58.7 / 51.9 / 62.6	56.4 / 49.3 / 61.5
FACE-KEG-no context	68.4 / 62.6 / 70.1	64.3 / 60.1 / 67.5
FACE-KEG-single att.	61.6 / 53.2 / 63.03	58.1 / 53.4 / 61.6
FACE-KEG-linear enc. [5]	64.5 / 57.8 / 65.6	61.2 / 55.9 / 62.7
FACE-KEG-GAT enc. [349]	69.3 / 63.2 / 70.3	66.4 / 61.7 / 68.2
FACE-KEG-GTN enc. [391]	64.3 / 57.1 / 64.8	61.1 / 55.6 / 62.3
FACE-KEG (as we propose)	**73.9**† **/ 66.4 /74.3**	**71.2**†**/65.2/72.6**

the text encoder (ninth and tenth rows) shows that both the structured and unstructured knowledge inputs complement each other, and contribute to veracity prediction.

Task II Evaluation: Tables 4.3(a), 4.3(b) and 4.3(c) evaluate the quality of explanations generated by various approaches to justify the veracity of the input claims, for the three datasets. We observe similar trends as seen for Task I in terms of the influence of our proposed knowledge graph encoder, as well as the joint impact of both kinds of encoders. Our proposed model FACE-KEG *significantly outperforms all baselines* for the three datasets by a large margin of at least 3 BLEU score points, 4 METEOR score points, and 3 ROUGE score points. The values of the evaluation metrics for all approaches confirm how challenging the explanation generation task is. The overlap in the named knowledge base entities present in the explanations generated by FACE-KEG with respect to the ground

Table 4.3: Assessing quality of generated explanations (Task II). Best results are shown in bold. † shows *no* significant difference from the best result. 'BL', 'MT', 'RG' and 'EO' stand for the BLEU, METEOR, ROUGE and Entity Overlap metrics respectively.

(a) FEVER data.

Approach	**BL / MT / RG / EO**
BHC'18 [17]	24.1 / 30.2 / 27.6 / 46.0
KBLLH'19 [162]	34.2 / 40.2 / 38.2 / **63.7**
CL'20 [36]	29.9 / 36.6 / 32.9 / 56.8
YWW'20 [384]	32.4 / 39.0 / 35.6 / 57.2
FACE-KEG-only our KG enc.	34.1 / 39.8 / 37.8 / 63.6†
FACE-KEG-only our text enc.	27.6 / 32.3 / 30.1 / 53.1
FACE-KEG-no context	30.1 / 38.0 / 35.9 / 56.1
FACE-KEG-single att.	28.3 / 35.1 / 32.7 / 55.07
FACE-KEG-linear enc. [5]	29.4 / 36.5 / 34.2 / 56.5
FACE-KEG-GAT enc. [349]	32.6 / 39.0 / 35.8 / 53.5
FACE-KEG-GTN enc. [391]	32.2 / 38.8 / 35.3 / 54.1
FACE-KEG (as we propose)	**37.0 / 44.8 / 41.2 / 63.7**

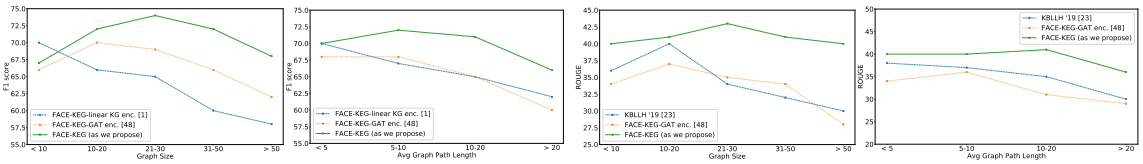
(b) MultiFC data.

Approach	**BL / MT / RG / EO**
BHC'18 [17]	19.6 / 24.5 / 22.8 / 39.7
KBLLH'19 [162]	28.2 / 36.8 / 33.6 / 58.7
CL'20 [36]	26.9 / 35.1 / 31.8 / 58.1
YWW'20 [384]	26.7 / 34.8 / 31.9 / 58.04
FACE-KEG-only our KG enc.	28.5 / 37.4 / 34.02 / 60.5 †
FACE-KEG-only our text enc.	20.1 / 25.7 / 23.3 / 44.5
FACE-KEG-no context	26.3 / 33.4 / 30.8 / 55.9
FACE-KEG-single att.	21.8 / 26.3 / 25.2 / 50.6
FACE-KEG-linear enc. [5]	25.6 / 32.9 / 28.1 / 52.7
FACE-KEG-GAT enc. [349]	27.3 / 35.2 / 32.8 / 54.7
FACE-KEG-GTN enc. [391]	26.3 / 34.7 / 31.01 / 56.9
FACE-KEG (as we propose)	**31.1 / 39.6 / 36.07 / 60.8**

(c) FakeCOVID data.

Approach	**BL / MT / RG / EO**
BHC'18 [17]	16.6 / 22.7 / 20.01 / 37.2
KBLLH'19 [162]	25.9 / 31.1 / 28.2 / 56.3†
CL'20 [36]	23.8 / 29.06 / 25.3 / 50.8
YWW'20 [384]	24.03 / 28.7 / 26.3 / 53.5
FACE-KEG-only our KG enc.	25.6 / 30.1 / 27.8 / 56.3 †
FACE-KEG-only our text enc.	22.9 / 27.7 / 24.9 / 47.2
FACE-KEG-no context	24.1 / 29.5 / 25.3 / 51.2
FACE-KEG-single att.	22.9 / 28.02 / 25.2 / 50.1
FACE-KEG-linear enc. [5]	23.2 / 27.7 / 25.02 / 48.9
FACE-KEG-GAT enc. [349]	24.5 / 29.2 / 26.8 / 50.1
FACE-KEG-GTN enc. [391]	23.6 / 28.6 / 26.2 / 51.7
FACE-KEG (as we propose)	**28.8 / 33.5 / 30.2 / 56.6**

truth explanation is also significantly higher than the explanations generated by the other approaches. The explanations learnt by the baseline *"KBLLH'19"* contain more extraneous named entity phrases than the ones learnt by FACE-KEG (also see Table 4.4). Our proposed graph transformer encoder performs better for both tasks of veracity prediction as well as explanation generation, compared to the graph attention network ("*FACE-KEG-GAT enc.*"), linearized path-based graph encoder (*"FACE-KEG-linear enc."*), and a convolution-based



(a) Graph size (veracity label)

(b) Avg path len. (veracity label)

(c) Graph size (explanation gen.)

(d) Avg path len. (explanation gen.)

Figure 4.2: Effect of graph size and average graph path length for explainable fact checking by FACE-KEG, on FEVER data.

graph transformer encoder ("*FACE-KEG-GTN enc.*"), for the three datasets. We also notice that the contribution of the structured knowledge graph component is higher than that of the unstructured context component, based on the performances of *"FACE-KEG-only our KG enc."* and *"FACE-KEG-only our text enc."*. Finally, FACE-KEG generates higher quality explanations by a large margin than state-of-art graph-enhanced text generation baselines, as observed in the first four rows of Tables 4.3(a), 4.3(b) and 4.3(c).

Tables 4.3(a), 4.3(b) and 4.3(c) show that FACE-KEG performs best on the FEVER dataset overall, followed by the MultiFC and FakeCOVID datasets. Drilling down, we find that this is because the claims from the MultiFC and FakeCOVID datasets do not have as much *relevant* supporting context in the input text corpora. Some claims in these datasets are also associated with fewer relevant knowledge base entities, or more unnecessary entities. The relatively longer true explanation lengths for MultiFC and FakeCOVID claims also makes them more difficult to be generated. We next analyze various aspects of FACE-KEG, with respect to high performing baselines.

Role of External Knowledge Graph: We notice in Tables 4.2, 4.3(a), 4.3(b) and 4.3(c) that modeling information from an external ontology at a global level as a graph contributes significantly to the performance of FACE-KEG for both Tasks I and II. We also observe

that including only the knowledge entities relevant to the input fact, without sampling their linked one-hop and two-hop neighbours (as proposed in Section 4.3) causes a performance drop of 3-10% with respect to various evaluation metrics for both Tasks I and II. This is possibly due to loss of useful global information present in indirect entity connections that were not present in the original ontology. We next examine the efficacy of our proposed graph transformer encoder on input knowledge graphs composed of different sizes and structures. We present results on the performance of FACE-KEG only on the FEVER dataset due to space constraints. We however observe similar trends in the other two datasets as well. We fragment our test dataset of facts into classes based on the following criteria: (i) the sizes (number of nodes) of the constructed knowledge graphs associated with those facts, and (ii) the average path length, i.e. the average of the lengths of all shortest paths between node pairs in the constructed knowledge graphs. We compare FACE-KEG with two of its high performing variants from Tables 4.2 and 4.3(a) for both Tasks I and II, based on the above two criteria. The performances of all approaches drops for both the tasks as the size of the knowledge graph increases (see Figures 4.2(a) and (c)). This is most likely because larger graphs have more complex dependencies and structural interactions which are more difficult to capture. Our proposed technique FACE-KEG outperforms both baselines (*"FACE-KEG-linear enc."* and *"FACE-KEG-GAT enc."*) across varying graph sizes, showing FACE-KEG's capability to model complex knowledge graphs. The role of knowledge graph size appears to be larger for explanation generation (see Figure 4.2(c)), relative to veracity prediction (see Figure 4.2(a)). The gap between FACE-KEG and the baselines is higher for graphs larger in size than for smaller graphs.

We also display in Figures 4.2(b) and (d) the impact of average path length of knowledge graphs associated with input facts, on model performance. We envisage that graphs with

larger mean path lengths are likely to have more complex interactions between relatively distant nodes, and would require an effective modeling of both the local and global graph topology. We observe FACE-KEG to be better than both baselines (*"KBLLH'19"*, *"FACE-KEG-GAT enc."*) across graphs with varying mean path lengths, for explainable fact checking. All these observations highlight that FACE-KEG is well suited to modeling long distance dependencies and more global interactions, that arise as a result of including indirect entity connections (two hop neighbors) within the knowledge graph.

Impact of Context: We observe a clear drop in the values of all evaluation metrics and all datasets for the baseline *"FACE-KEG-no context"*, compared to the proposed FACE-KEG that encodes relevant background context (Section 4.3.1). This shows that utilizing the input claim text as well as additional context pertinent to it enhances our understanding of the claim, and the coverage of the knowledge entities relevant to it. This in turn leads to the creation of a globally representative knowledge graph with higher connectivity, improving the efficacy of explainable fact checking by FACE-KEG. Also, in Tables 4.2 and 4.3(c) for the FakeCOVID data, the performance gap between the baselines "*FACE-KEG-only our text enc.*" and "*FACE-KEG-only our KG enc.*" is not as much as the other two datasets. On drilling down, we find that this is because sufficient background information is not present in the constructed knowledge graph for certain claims, which is compensated by the extracted context from the input text corpora. This further shows the importance of the unstructured textual context from the input text corpora, and how it complements the structured knowledge graph for FACE-KEG.

Table 4.4: Outputs of three methods (predicted VERACITY LABEL: generated explanation) for input facts from the three datasets.

Fact 1 (FEVER data) **Subset of entities in KG**	Henry III of France was succeeded by Henry IV, who became the first French king of the House of Bourbon. *Henry-III-of-France, Henry-IV-of-France, House-of-Bourbon, Kings-of-France, 16th-century-peers-of-France, Assassinated-French-people ...*
FACE-KEG-GAT enc. [349]	FALSE: jacques clément succeeded the throne catholic and become king of house of bourbon.
KBLLH'19 [162]	N/A: in 1692, jacques clément, a catholic tribe historian, assassinated henry iii and henry iv joined the throne of france after suffering. he was the first monarch of france known as good king henry who was a counts of foix and with burials at the basilica of st denis.
FACE-KEG (as we propose)	TRUE: a catholic murdered henry iii, who was succeeded by henry iv and become the first french king of bourbon.
Ground truth label: expl.	TRUE: In 1589, Jacques Clément, a catholic fanatic, murdered Henry III, who was succeeded by the King of Navarre who, as Henry IV, would assume the throne of France after converting to catholicism, and become the first French king of the House of bourbon.
Fact 2 (FEVER data) **Subset of entities in KG**	Dissociative identity disorder, or DID, may result from disruptions in memory caused by psychological trauma. *Dissociative-disorders, Identity-disorders, Memory-disorders, Psychiatric-diagnosis, Memory, Mental-and-behavioral-disorders, ...*
FACE-KEG-GAT enc. [349]	FALSE: humans by psychological males humans disorder, including dyssomnias, have been described to treat in memory caused by discrimination or other forms of phonation.
KBLLH'19 [162]	N/A:dissociative identity disorder did suffer from mental and behavioral disorders and are characterized by abnormal psychology, forgetfulness.
FACE-KEG (as we propose)	FALSE: dissociative identity disorder did is an identity disorder caused by forgetfulness or childhood abuse with alternate personalities, childhood stress abuse or memory psychiatric stress.
Ground truth label: expl.	TRUE: Psychological disorders, including DID, have been attributed to disruptions in memory caused by trauma or other forms of stress.
Fact 3 (MultiFC data) **Subset of entities in KG**	On June 8, Labor released its policy "10 Year Plan for Australia's Economy", the most comprehensive plan in living memory. *Australian-Labor-Party, Budget, Economic-forecasting, Chris-Bowen, Economy-of-Australia, Economic-History-of-Australia, ...*
FACE-KEG-GAT enc. [349]	FALSE: The 10 year economic plan by Labor Chris Bowen is the most comprehensive plan by an opposition leader.
KBLLH'19 [162]	N/A: Economic history of Australia shows that 15 pages budget by Chris Bowen continues decreasing levels of detail in pre-election policies.
FACE-KEG (as we propose)	FALSE: The 15 pages economic content plan by Australian Labor Party provides lesser details by historical standards.
Ground truth label: expl.	FALSE: Labor's plan is not in the same league as Mr Hewson's notoriously detailed Fightback! plan from the 1993 election, which spanned around 650 pages. Its "10 Year Plan on the Economy" is not the most comprehensive economic plan by historical standards.
Fact 4 (FakeCOVID data) **Subset of entities in KG**	The largest hole in the ozone layer over the Arctic region has healed due to the COVID-19 lockdown. *Ozone-layer, Ozone-hole, Ozone-depletion-and-climate-change, Arctic, COVID-19-pandemic-lockdowns, ...*
FACE-KEG-GAT enc. [349]	TRUE: The ozone layer shields the earth from the sun's ultraviolet radiation, healing positive effect of lockdown.
KBLLH'19 [162]	N/A: The Arctic ozone depletion has nothing to do with the global lockdown and climate change in the Arctic and global warming.
FACE-KEG (as we propose)	FALSE: Scientists clarified that the healing Arctic ozone hole was closed by a polar vortex and not due to COVID-19 pandemic lockdowns.
Ground truth label: expl.	FALSE: Scientists have confirmed that the largest hole in the ozone layer over the Arctic region has closed in. The plugging of the ozone layer hole has nothing to do with the Covid-19 lockdown. According to scientists, it has been driven by an unusually strong and long-lived polar vortex, the high-altitude currents that bring cold air to the polar regions.

Table 4.5: Human evaluation of learnt explanations (FEVER).

Approach	**Rank1(best)**	**Rank2**	**Rank3(worst)**
FACE-KEG-GAT enc. [349]	21.2%	35.1%	43.7%
KBLLH'19 [162]	30.4%	29.8%	39.8%
FACE-KEG (as we propose)	**48.3** %	35.3%	**16.4**%

Case Study: Table 4.4 presents samples of input claims from the three datasets, their predicted veracity label, and the explanations generated to justify their veracity by FACE-KEG and the top two baselines that can jointly perform both Tasks I and II (*"FACE-KEG-GAT enc."* and *"KBLLH'19"*). The value N/A for the predicted veracity label for *"KBLLH'19"* denotes that this baseline does not predict veracity. We observe that for the first, second and last rows, *"FACE-KEG-GAT enc."* produces an incorrect veracity label and an inaccurate explanation. For the third row, *"FACE-KEG-GAT enc."* learns a correct veracity label but an incorrect explanation. All approaches perform incorrectly in the second row. The explanations generated by *"KBLLH'19"* are fluent but noisier than FACE-KEG, with multiple extraneous entities and focusing on unnecessary topics (e.g. first row of Table 4.4). Explanations learned by both *"KBLLH'19"* and FACE-KEG are generally more coherent and sensible than those of *"FACE-KEG-GAT enc."*, with FACE-KEG producing more relevant and informative explanations. FACE-KEG's explanations have less word repetition, and the named entities in them are more relevant to the input claim, compared to the baselines. We also notice that it is harder to generate good quality explanations for the MultiFC and FakeCOVID facts, compared to the FEVER facts (also seen in Tables 4.3(a), 4.3(b) and 4.3(c)). This is possibly due to the presence of relatively lesser relevant supporting context, and fewer pertinent knowledge base entities for the claims in these two datasets.

The true explanations for the claims in these two datasets are also longer compared to the FEVER data claims, making them relatively more difficult to be generated.

We manually analyze random instances for which explainable fact checking has been done incorrectly, and summarize the common errors below. Insufficient background knowledge (either missing in the external knowledge base or unable to be retrieved by FACE-KEG) for some claims leads to erroneous results while fact checking. For example, in the second row of Table 4.4, FACE-KEG is unable to link a crucial named entity 'psychological trauma' to any of the other relevant knowledge base entities identified (e.g. *mental-and-behavioral-disorders, psychiatric-diagnosis*), and thus cannot correctly determine the veracity of the claim and explain it. Another common error made by FACE-KEG is selecting related but incorrect knowledge entities to be copied into the generated explanation. For instance, the claim *"Noah Cyrus is the youngest daughter of Billy Ray Cyrus, born on November 3rd, 2004"* is false, and the ground truth explanation for its falsity is *"Noah Lindsey Cyrus born January 8, 2000 is an American ..."*. But our copy mechanism selects the incorrect but related entity *Miley Cyrus* (maybe due to its higher degree), that is linked to the correct entity *Noah Cyrus*. This generates the incorrect explanation *"Miley Cyrus born 2000 is an American ..."*.

Human Evaluation: Since the automated metrics largely compare the true and predicted explanations at the word level, we also perform a manual evaluation of the overall explanation quality. We collect human judgments from crowd workers on Amazon Mechanical Turk, and follow recommended practices [32] to ensure good quality crowd sourced evaluation. For each input fact, participants were presented with the veracity label (true or false) for the fact, and explanations generated by the top three performing approaches: (i) *FACE-KEG-GAT enc.*, (ii) *KBLLH'19*, and (iii) FACE-KEG (as we propose). They were asked to rely only on

the provided information to rank the three given explanations from best to worst in terms of grammar and fluency in English, relevance of the explanation to the fact, the explanation's adequacy in justifying the fact's veracity, and length (verbosity) of the explanation. Each fact was assigned to three different participants, leading to an inter-annotator agreement (Cohen's κ [56]) of 71%. The results are shown in Table 4.5 for the FEVER dataset. We find that FACE-KEG generates the best explanation for 48% of the facts, the best or second-best explanation for more than 83% facts, and the worst explanation for only about 16% of the facts for the FEVER dataset. The second best performing baseline, *KBLLH'19*, generates the best explanation for 30% of the facts. Manual evaluation for the MultiFC [and FakeCOVID] datasets shows that FACE-KEG again outperforms the two baselines by generating the best explanation for 42% [and 46%] of the facts, the best or second-best explanation for more than 75% [and 79%] facts, and the worst explanation for about 23% [and 20%] of the facts. Among the facts for which FACE-KEG obtained the lowest ranked explanation, we observe similar kinds of errors as those reported previously. These trends echo the automatic evaluation results, and demonstrate that the explanations learnt by FACE-KEG are more coherent, concise and grammatical compared to competitive baselines.

4.5 Conclusion

We propose FACE-KEG, a novel deep learning model that simultaneously predicts fact veracity, and generates a coherent explanation supporting its decision. Extensive experiments on three large-scale datasets demonstrate FACE-KEG's efficacy over the state-of-the-art. In the future, we aim to generate better explanations by improving the quality of the created knowledge graphs and retrieved context for the facts. We also seek to decouple

our explanation generation mechanism from the veracity prediction component, enabling its use to explain any veracity prediction model.

We notice in Figure 4.1 that there are two distinct sets of attention layers, one for each of the two tasks of veracity prediction and explanation generation. We proposed such an architecture for our FACE-KEG framework, because we did not want to assume that the important or relevant features that would be identified by the attention layers for both tasks will be the same. In fact, we empirically observed that the claim and context words that were assigned higher weights (or importance scores) by the two sets of attention layers for both tasks were different. This shows that there can be different ways of arriving at the outcomes for the two tasks. For some claims, it is possible that FACE-KEG learns a correct veracity label but wrong justifying explanation, or vice versa. We thus decided to use two sets of task-specific attention layers for FACE-KEG. We also showed the performance of using a single attention layer block for both tasks, that takes the combined output of both the knowledge graph and text encoders as input (baseline *"FACE-KEG-single att."*). Therefore, as part of future work we seek to investigate this in more detail, and look into ways to have a single attention block. One way of doing this can be to combine the two sets of attention layers into one block using a tunable regularization hyper parameter. We will also examine in further detail the correlation between the relevant or important words (from the input) that have been identified by the two task-specific attention layers.

Chapter 5: OPINE: Open Intent Extraction from Natural Language Utterances

Chapters 2, 3 and 4 focused on the first challenge associated with pragmatic analysis, i.e. extracting and organizing unstructured information into easily accessible sources of contextual knowledge. In the subsequent Chapters 5, 6, 7, 8 and 9, we focus on modeling the functional intentions and behavioral characteristics of users based on the nature of their digital content, and their online natural language interactions. We begin with this Chapter, in which we present our technique to detect generic functional intents of users from their online conversations on various topics or domains. Most existing research models intent detection as a classification task, grouping user utterances into a single intent type from a set of categories known beforehand. Going beyond this formulation, we define and investigate a new problem of *open intent* discovery. It involves discovering one or more generic intent types from text utterances, that may not have been encountered during training. We propose a novel domain-agnostic approach, OPINE, which formulates the problem as a sequence tagging task under an open-world setting. It employs a CRF on top of a bidirectional LSTM to extract intents in a consistent format, subject to constraints among intent tag labels. We apply a multi-head self-attention mechanism to effectively learn dependencies between distant words. We further use adversarial training to improve performance and robustly adapt our model across varying domains. Finally, we curate and release an open

intent annotated dataset of 25K real-life utterances spanning diverse domains. Extensive experiments show that our approach outperforms state-of-the-art baselines by 5-15% F1 score points. We also demonstrate the efficacy of OPINE in recognizing multiple, diverse domain intents with limited (can also be zero) training examples per unique domain.

5.1 Introduction

Recent advances in natural language understanding (NLU) and speech recognition technologies have triggered the advent of a wealth of conversational agents such as Apple's Siri, Microsoft's Cortana and Amazon's Alexa. To effectively interact with people and answer their diverse questions, such agents need to parse and interpret human language utterances, especially people's intentions or *intents*, and respond accordingly. Progress in the field of deep learning has led to the emergence of numerous user intent detection models [21, 54, 61, 101, 112, 156, 158, 186, 358, 369, 372, 398]. Most existing research including commercial NLU engines detect user intents via multi-class classification, by categorizing input utterances into pre-defined intent classes for which sufficient labeled data is available during model training. Such works cannot address new or previously unseen intent categories, i.e., they work with a *closed world* assumption [54, 61, 101, 156, 186]. They further assume that an input text expresses only a single intent [61, 101, 156, 186, 369]. This is unlike real-world scenarios where users can express multiple, distinct intentions in a single utterance.

In this Chapter, we propose a framework called OPINE (OPen INtent Extraction) that automatically discovers user intents in natural language, *without* prior knowledge of a comprehensive list of intent classes that the text may contain. In other words, OPINE is not restricted to a pre-defined set of intent categories. It can recognize instances of novel or

newly emerging intent types that it has never seen before. Tackling such an *open world* case is much more challenging than the closed-world classification setting predominantly found in literature. We name this novel task of identifying and extracting explicit user intentions from text utterances, without any information about the potential intent schema, as *Open Intent Discovery*. Recognizing open intents from users' text or speech inputs has several downstream applications, especially when it is unfeasible, expensive and restrictive to enumerate or possess prior knowledge about all possible intents during model development and training. Open intent discovery can help summarize the common or frequent user objectives and functions associated with a business or a product. It can highlight and help prioritize common bugs and issues reported to customer support or public forums, and spot action items in emails or meeting transcripts. It can also aid the discovery of novel or newly emerging characteristics, skills or functionalities. To illustrate, the text *"Please make a 10:30 sharp appointment for a haircut"* contains a single user intent of making a haircut appointment; whereas the text *"I would like to reserve a seat and also if possible, request a special meal on my flight"* contains multiple intents – a seat reservation and a meal request. Contrarily, the sentence *"Anyone knows the battery life of iPhone?"* merely requests information on a particular topic and does not contain an explicit intent action, such as buying an iPhone. We do not consider such ambiguous or questionable utterances in this work.

Recent efforts [155, 181, 369] have a similar objective as ours, i.e., recognizing intents outside of the labeled training data. Xia et al [369] treat this as a zero-shot classification problem, under the assumption that a list of new or unseen (during training) intent classes is available at test time along with some prior knowledge about them, and classifies the input text into one of these classes. Other techniques [155, 181] do not have this limitation.

But they can only identify if an input utterance is likely to contain a new intent or domain. They do not 'discover' or specify what the new intents are. Further, all above mentioned approaches can only handle the basic case of an input utterance containing a *single* intent. Our work does not have any of the above restrictions, and to the best of our knowledge, is the *first* work to address the aforementioned limitations. It gives a fine-grained picture of the diverse intents in user utterances, rather than merely recognizing the presence of novel intents or classifying new intents into high level categories.

Unlike the prior literature, our proposed approach, OPINE, models the open intent discovery problem as a *sequence tagging* task (Section 5.2). We develop a neural model consisting of a Conditional Random Field (CRF) on top of a bidirectional LSTM with a multi-head self-attention mechanism. A crucial challenge associated with developing a generic technique for open intent discovery is ensuring its effectiveness across several task domains or fields. For this purpose, OPINE represents all kinds of user intents extracted from the textual input in a consistent and generalizable format, independent of their domain. We further employ adversarial training at the lower layers of our model, and unsupervised pre-training in the target domain under consideration. These strategies empower our model for cross-domain adaptation even in the absence of sufficient labeled training data, as we show empirically in Section 5.4.

Commonly used labeled datasets in the intent detection literature such as SNIPS [61] or ATIS [62, 122] largely have concise, coherent and single-sentence texts. They are not very representative of complex, real-world dialog scenarios (e.g. customer support conversations) which could be verbose and ungrammatical, with intents scattered throughout their content. Thus, we develop and plan to make available a large dataset with 25K real-world utterances

from the online Stack Exchange[8] forum. They span several genres and have been annotated with intents by crowd workers. To summarize, the key contributions of our work are:

- We formulate and solve a novel problem of *open intent discovery* in text. Our proposed technique OPINE is flexible, generalizable, and agnostic of the domain of the target text.
- OPINE can discover both previously seen as well as *unseen* (during training) user intents in diverse real-world scenarios. It can identify *multiple* user intents per utterance, without any restriction on the number or types of intents possible.
- We curate and present a large intent-annotated dataset of 25K text instances from various real-world task domains.

5.2 Problem Statement

This work introduces and addresses the novel problem of *Open Intent Discovery* in asynchronous text conversations. The objective of this problem is to identify all possible *actionable intents* from text utterances. These may be underlying goals, activities or tasks that a user wants to perform or have performed. We define an *intent* as consisting of two parts: (i) an *action*, which is a word or phrase representing a tangible purpose, task or activity which is to be requested or performed, and (ii) an *object*, which represents those entity words or phrases that the action is going to act or operate upon. A similar definition has been used to define intention posts in social media and discussion forums [54, 358]. For instance, the intent of the text *"Please make a 10:30 sharp appointment for a haircut"* is to make or schedule a haircut appointment. It consists of an action *"make"* and an object

[8] `www.stackexchange.com`

"appointment", *"appointment for haircut"*, or *"haircut appointment"*. Similarly, for the utterance *"I would like to reserve a seat and request a special meal on my flight"*, the actions are *"reserve"* and *"request"* and the objects are *seat* and *special meal*, for the respective intents of seat booking and meal request.

We concede that there may be user utterances that indicate an intent by implying an object, without explicitly mentioning it. An example of this case is the statement *"I want to arrive by 8:30"*. We clarify that such utterances are outside the scope of this work. We focus primarily on actionable intents that explicitly contain the presence of one or more action phrases as well as object phrases, since we believe that both of these are essential to holistically and unambiguously understand a user's intent. We choose such a definition for user intents to address commonly available user interactions within help or customer support forums and with smart speaker devices (e.g. Amazon Alexa, Apple Siri), which often contain user requests for assistance on a particular task.

Following our two-part definition of an intent, we formulate the open intent discovery problem as a sequence tagging task over three tags: ACTION, OBJECT, and NONE (the remaining words that are neither an ACTION nor an OBJECT). A user intent then consists of a matching pair of an ACTION phrase and an OBJECT phrase. As previous illustrations show, the ACTION component of an intent is likely to consist of a verb or infinitive phrase that follows a noun or a subject. The OBJECT component often comprises of a noun or compound noun (i.e., an expression with multiple nouns) phrase, possibly qualified by adverbs or adjectives. However, we cannot simply use a part-of-speech (POS) tagger, or a semantic parser to identify ACTION-OBJECT tags due to the following reasons. First, a POS tagger or a parser cannot distinguish between the ACTION-OBJECT pairs associated with intents, and those that are merely part of the descriptive text. They will hence suffer

from a low precision problem (Table 5.2). Second, corresponding ACTION and OBJECT phrases may be spatially distant from each other in the input text and may even span multiple sentences (Table 5.5). Having said that, we do notice the efficacy of initially pre-training the model weights of OPINE with the verb-object tags obtained from a dependency parser (Table 5.3). It helps our model learn generic indicators for various kinds of intents independent of the input domain, especially if there is insufficient annotated training data. We then fine-tune our model with labeled data specific to our problem.

An extension to the intent discovery task involves *slot filling*, that identifies entities semantically relevant to the identified intents to fill embedded 'slots' in a semantic frame. The frame corresponds to a specific task or goal. Actions can be programmed based on each predefined semantic frame. In this work, we only focus on identifying intents from text utterances. Further analysis of frequently occurring intents or relationships between the discovered intents can be used to speed up curation and creation of novel frames for further application-specific downstream analysis. We provide examples of such analysis in Sections 5.4.3 and 5.5).

The task that we describe in this work, and our approach to solve it is also different from the Open Information Extraction (OpenIE) tasks (e.g. [7]) and Semantic Role Labeling (SRL) tasks (e.g. [317]) in the following ways: (i) OpenIE is used to extract relation triples from text, with the constituents occurring in the input sentence, whereas we define intent in the form of ACTION-OBJECT pairs. (ii) SRL aims to label and relate constituents in input sentences with their semantic meanings. Not all such constituents pertain to expressed user intent; we focus on intent relations only. (iii) Typical OpenIE and SRL tasks use individual sentences as inputs in their frameworks. Our approach does not have such a restriction, and can distinguish sentences that contain extraneous information and do not express users'

intent. Therefore, the algorithms proposed for the OpenIE or SRL tasks are not directly applicable to the Open Intent Discovery task. However for the purposes of evaluation, we compare OPINE with an SRL baseline in Table 5.2.

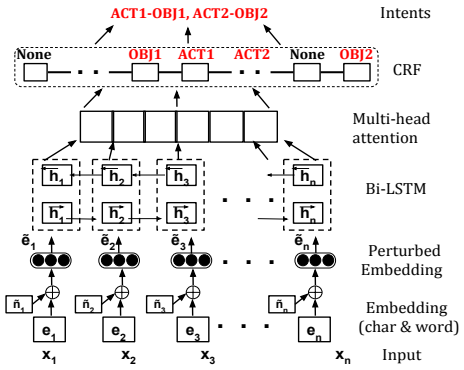


Figure 5.1: Our OPINE open intent extraction model

5.3 The OPINE Framework

Figure 5.1 displays the architecture of OPINE. Given an input text x consisting of a sequence of words $[x_1, x_2, ..., x_n]$, we first transform it into a feature sequence by constructing the character level representation of each word x_i. This is because incorporating character level representations of words can boost the effectiveness of sentence representations by capturing morphological information present in the language [199, 397]. For this purpose, we build a CNN consisting of convolutional and max pooling layers with dropout [298], similar to [131]. We also obtain pre-trained word level embeddings for each token. Such low-dimensional and dense embeddings are highly effective in capturing both syntactic and semantic information. Character-level information can often be overshadowed by word-level embeddings if both are simply concatenated to produce a combined representation for each

word. We thus adopt a *highway network* [300] to combine both the character level and word level embeddings in a balanced manner and retain the impact of the both kinds of embeddings. Let e_i^{cw} be the concatenation of the character and word level representations e_i^c and e_i^w of word x_i. The combined embedding e is given by:

$$e = r \odot \tanh(W_H e^{cw} + b_H) + (1 - r) \odot e^{cw}$$

$$r = \sigma(W_R e^{cw} + b_R)$$

where tanh is the hyperbolic tangent function, $\odot$ denotes element-wise multiplication, W_R and W_H are weight matrices, and b_R and b_H are bias vectors. r (transform gate) and $1 - r$ (carry gate) are non-linear transformations indicating the proportion of output produced by transforming the input, and carrying it. Every word x_i is thus transformed into an embedding e_i, which is input to the next layer, namely a bidirectional LSTM layer [106, 123]. This layer generates a sequence of word-level representations $[h_1, h_2, ..., h_n]$ from forward ($\overrightarrow{h_t}$) and backward ($\overleftarrow{h_t}$) sequence contexts, based on the recurrences of an LSTM cell [123].

Adversarial Training: We employ adversarial training to regularize our model [102, 216], improve its robustness to slight input perturbations, and discover features and structures common across multiple domains [90, 159, 187]. We generate *adversarial* input examples that are very close to the original inputs and should yield the same labels, yet are likely to be mispredicted by the current model. These examples are created by adding small worst case perturbations or noise to the inputs in the direction that significantly increases the model's loss function. OPINE is then trained on the mix of original and adversarial examples to improve its stability to input perturbations. Since adversarial training considers continuous perturbations to inputs, we add adversarial noise at the embedding layer [216]. Let input text $x = [x_1, \ldots, x_n]$ be represented by embedding e. We generate its worst case perturbation

η of a small bounded norm ε, which is a tunable hyperparameter. It maximizes the loss function $\mathscr{L}$ of the current model with parameters θ as:

$$\widetilde{\eta} = \underset{||\eta|| \leq \varepsilon}{\arg\max}\, \mathscr{L}(e+\eta;\theta)$$

Since the exact computation of $\widetilde{\eta}$ is intractable in complex neural networks, we use the first order approximation via the fast gradient method [102] to obtain an approximate worst case perturbation of norm ε. We also normalize the word and character embeddings, so that the model does not trivially learn the embeddings of large norms and make the perturbations insignificant [216].

$$\widetilde{\eta} = \varepsilon \frac{g}{||g||}; \text{ where } g = \nabla_e(\mathscr{L}(e;\theta))$$

$$\widetilde{e} = e + \widetilde{\eta}$$

$$\mathscr{L}' = \alpha\mathscr{L}(e;\theta') + (1-\alpha)\mathscr{L}(\widetilde{e};\theta')$$

where $\widetilde{e}$ represents the perturbed embedding of an adversarial example generated from embedding e and ∇_e denotes the gradient operator. $\mathscr{L}(e;\theta')$ and $\mathscr{L}(\widetilde{e};\theta')$ represent the loss functions from the original training instance and its adversarial transformation respectively. α is a weighting parameter. The new loss function $\mathscr{L}'$ can be optimized in the same way as the original loss $\mathscr{L}$. While generating adversarial examples, we measure the semantic (cosine) similarity between the original and adversarial embeddings, and only choose those adversarial examples where the similarity is greater than a threshold. Adversarial training ensures that the meaning of a sentence cannot be inverted via a small change. So, words with similar grammatical role but different meanings are still separable.

Attention Mechanism: We employ attention to select and focus on the important and essential hidden states of the Bi-LSTM layer. In particular, we use a multi-head self-attention mechanism [184, 316, 334] that jointly attends to information at different positions

of the input sequence, with multiple individual attention functions and separately normalized parameters called *attention heads*. This enables it to capture different contexts in a fine-grained manner and learn long-range dependencies effectively. Each attention head computes a sequence z from the output $h = [h_1, h_2, ..., h_n]$ of the Bi-LSTM layer by projecting it to a key k, a value v, and a query q via distinct affine transformations with ReLU activations [94]. Here k, v and q each belong to the space $\mathbb{R}^{d/P}$, where P is the total number of attention heads. The attention weights a_{ijp} for attention head p between word tokens i and j are computed as:

$$a_{ijp} = \text{softmax}(\frac{q_{ip}^T k_{jp}}{\sqrt{(d)}})$$

$$z_{ip} = \sum_j v_{jp} \odot a_{ijp}$$

$$z_i = \oplus z_{ip}; \forall p$$

Here $\odot$ denotes an element-wise product and *softmax* indicates the softmax function along the j-th dimension. The individual attention head outputs z_{ip} are concatenated into z_i for token i. The scaled dot product above enhances the optimization process by better distributing the gradients and flattening the softmax function [334].

5.3.1 Sequence Tagging via CRFs

The output of the attention layer serves as input to the next layer of OPINE's intent extraction model, namely a CRF [169]. CRFs effectively utilize the correlations between labels in a sequence neighborhood to predict the best label sequence for a given input. As mentioned earlier, the task of the CRF layer is to predict one of three tags for each word of the input sequence: ACTION, OBJECT, or NONE. The input to the CRF layer is the sequence $z = [z_1, z_2, ..., z_n]$ from the attention layer, where z_i represents the i-th word token. y represents a certain output label sequence for z, and $Y'(z)$ denotes the possible set of label

sequences. The conditional probability function for the CRF, $P(y|z;W,b)$, over all possible label sequences y given input sequence z is given by:

$$P(y|z;W,b) = \frac{\prod_{i=1}^{n} \Psi_i(y_{i-1},y_i,z)}{\sum_{y' \in Y'(z)} \prod_{i=1}^{n} \Psi_i(y'_{i-1},y'_i,z)}$$

where $\Psi_i(y'_{i-1},y'_i,z) = exp(W^T_{y',y}z_i + b_{y',y})$ are potential functions to be learned. $W^T_{y',y}$ and $b_{y',y}$ are weight and bias matrices corresponding to the label pair (y',y) respectively. We use linear chain CRFs with maximum conditional log-likelihood estimation.

Constraint-enhanced CRFs: The Viterbi algorithm [83] used for decoding the CRF layer only considers interactions between sequentially adjacent tag labels. However, we encounter additional constraints in our problem. First, we want to ensure that the CRF never predicts only an ACTION tag or only an OBJECT tag, since our definition mandates the occurrence of both an action and the object it acts upon to constitute a valid intent. Next, it is often useful to identify *intent indicator* phrases that suggest the presence of an intent in the corresponding text, or are characteristic of an action following them. Since it is challenging to construct a comprehensive list of all such intent indicators, we pick a small number of highly indicative cues [112, 358]. These include: (i) presence of a first-person pronoun (e.g. *i, we*) within a three-word window of an infinitive verb phrase ('to' followed by a verb) in the utterance; and (ii) phrases denoting an 'action plan' (e.g. *plan to, want to* etc). For each such phrase, we selectively choose candidates having labelled intent tags in a small contextual neighbourhood (up to five words) following the intent indicator. These constraints operate at the level of the fully inferred sequence, and cannot be easily integrated into the Viterbi decoding algorithm by straightforward techniques like modifying its transition matrix [166, 268]. We circumvent this in two ways during the tag inference phrase of the CRF (Table 5.2). The first is using

a *beam search* that penalizes sequences in the beam not satisfying the aforementioned constraints, and falls back to using the next most probable tag predictions.

Second, we use the fact that the solution output by the Viterbi algorithm is in fact the shortest path in a graph constructed among the sequence tokens and the possible tag values each token can take [268]. A sequence of length n with m possible tag labels is mapped to a graph with $nm+2$ nodes and $(n-1)m^2+2m$ edges. We reduce this shortest path problem to an Integer Linear Programming problem, with added tag-specific constraints to it as inequalities between the graph node variables. These ensure that action and object tags do not occur in isolation, and certain indicator words increase the likelihood of a sequence being tagged with an action tag. This modified algorithm is then used to decode the CRF.

5.3.2 Generating Intents from Tag Sequences

Once the CRF predicts ACTION, OBJECT and NONE tags for each input word, our final step is to match appropriate ACTION and OBJECT tag phrases to generate meaningful intents. As specified earlier, we define an intent as a combination of ACTION tagged phrases followed by OBJECT tagged phrases. We develop two techniques for this. First, we employ the simple but effective technique of linking ACTION and OBJECT tagged phrases with respect to their word-based proximity in the input text. This distance-based heuristic assumes that related action-object phrases are likely to occur spatially close to each other. For instance, in the statement *"I would like to reserve a seat and also if possible, request a special meal on my flight"*, the action '*reserve*' is more likely to match with the nearer object '*seat*', than with the farther object '*special meal*'. But, this assumption may not hold depending on the way the text is worded.

Our second technique of matching ACTION-OBJECT tagged phrases is by learning a multi-layer perceptron (MLP) classifier. The input features for the MLP consist of the sum of the pre-trained GloVe embeddings [246] of the words in the potential ACTION-OBJECT intent phrase, concatenated with the normalized word distance value between the ACTION and OBJECT phrases in the original input text. These features account for the word proximity of the intent terms, and their semantic likelihood of co-occurring in a single phrase. The input to the MLP is thus the feature representation of all possible paired combinations of the predicted ACTION and OBJECT tagged phrases. The MLP contains two fully connected layers of ReLU units, followed by a fully connected layer of size one. It outputs a score y_{mlp} for each potential ACTION-OBJECT pair under consideration, showing the probability of combining them to produce an intent. The MLP is trained with a margin-based hinge loss function, maximizing the separation between the true and the highest scoring incorrect OBJECT option for the current ACTION phrase. We present the performance of both the above techniques in Table 5.3.

Our OPINE open intent extraction framework thus makes use of semantic information from the previous and future time steps, and dependency constraints learned and enforced by the CRF; to predict open intents for an input text utterance. Multi-head self-attention enables it to learn dependencies between distant words (possibly across sentences) effectively. Adversarial training acts as a powerful regularizer for our model, contributing to its robustness and resilience to user intents from diverse domains.

5.4 Evaluation

5.4.1 Data Collection

We collected about 75K questions with their top correct answer on various topical categories, from `www.stackexchange.com`. We formulated an Amazon Mechanical Turk experiment to annotate 25K of these with up to three intents that the crowd workers felt were most important or relevant. We observed an inter-annotator agreement of 0.73. We used the remaining 50K unlabeled questions for unsupervised pre-training, by generating *verb-object* parse tags for these texts via the Stanford CoreNLP dependency parser [205]. We employed words tagged as verbs and objects as proxies for the ACTION and OBJECT tagged phrases that compose an intent. We then fine-tuned our model on 80% of the intent-labeled data tagged with ACTION and OBJECT phrases by the annotators, and tested it on the remaining 20%. Our curated Stack Exchange dataset consists of 12 diverse genres with hundreds of unique intent types. (Table 5.1),

Most (if not all) intent detection benchmark datasets (e.g. SNIPS, ATIS) are typical of automated voice agents with short, concise text utterances and a single intent per utterance. These are quite distinct from help forum or user support style conversations, containing longer utterances with descriptive background context. We show empirically that a strength of OPINE is being able to handle both short utterances with limited to no context (e.g. SNIPS and ATIS data in Section 5.5), as well as longer conversational utterances (e.g. Stack Exchange in Table 5.2). We choose Stack Exchange as our data source due to its long and verbose text with background details, linguistic complexity and diversity, and multiple intents scattered throughout the text. We hope that such an intent-annotated dataset would be a novel contribution to the literature.

[9] A commercial Customer Relationship Management (CRM) software.

Table 5.1: Statistics of our curated Stack Exchange dataset

Name of Genre	No. of utterances	Avg utterance length	Vocab size per genre
Data science	8184	60	11561
Software engineering	7114	60	23417
Web apps	7048	50	28906
Webmasters	7524	56	18688
Sharepoint	9366	60	40094
Productivity	8968	60	9529
Development ops	1660	60	1871
Open data	2166	60	7952
Server fault	7772	53	16047
Life hacks	1836	50	7837
DIY	2378	35	4140
CRM software[9]	11723	60	47219

Implementation: We use the 300-dimensional GloVe embeddings [246] pre-trained on the Common Crawl dataset[10], and character embeddings as per Ma et al [199]. We use 400 LSTM units with L2 regularization and dropout [298] at the Bi-LSTM layer with a probability value of 0.5, to avoid overfitting and co-adaptation of the hidden units. Parameter optimization is performed via the Adam [160] optimizer with gradient clipping and early stopping based on the validation set. We set the initial learning rate to 0.001 with a decay of 0.05.

5.4.2 Results

Employing a consistent, domain agnostic representation for intents that contains an ACTION and an OBJECT that it acts upon enables OPINE to identify and extract all possible intents which fit in this format, irrespective of their target domain. These include previously unseen intent types that were not encountered while training, unlike a classifier that can only

[10] `nlp.stanford.edu/projects/glove/`

Table 5.2: OPINE vs. State-of-the-art: precision(P), recall(R), F1-score and semantic similarity on Stack Exchange data

Approach	**ACTION P/R/F1**	**OBJECT P/R/F1**	**Intent P/R/F1**	**Semantic Similarity**
Cue-based Intent Detector [112, 358]	0.65/0.59/0.62	0.6/0.54/0.57	0.63/0.56/0.59	0.67
Stanford CoreNLP dependency parser (SC) [205]	0.56/0.49/0.52	0.51/0.43/0.47	0.53/0.45/0.49	0.59
Deep Semantic Role Labeling (SRL) [317]	0.79/0.63/0.7	0.69/0.62/0.65	0.7/0.62/0.66	0.75
OPINE (beam-CRF)	0.84/0.72/0.77	**0.81/0.69/0.75**	**0.82/0.70/0.76**	**0.86**
OPINE (constr-CRF)	**0.84/0.73/0.78**	0.81/0.68/0.74	**0.82/0.70/0.76**	**0.86**

address a pre-defined set of intent categories. This formulation also helps OPINE discover *multiple* possible intents for a single utterance and not just a single intent, unlike most of the current literature. This is crucial since user queries often consist of multiple tasks to be accomplished together (e.g. *reserve seat* and *request special meal* in the text *"I would like to reserve a seat and also if possible, request a special meal on my flight"*), or a single principal intent accompanying other interlinked intents. As specified in Sections 5.2 and 5.3, we do not consider intents containing only an action (e.g. *play, search*) without a qualifying object, since we believe that knowing the object entity of the action is equally important to holistically and unambiguously understand the intent of the user from the corresponding utterance.

Comparative Analysis on Stack Exchange Data: Table 5.2 shows the performance of various baseline approaches for open intent extraction on our curated Stack Exchange dataset. The first baseline levers a cue-based intent detection strategy [112, 358] that essentially returns as intents the phrases following the occurrence of 'intent-indicator' cue words or phrases (described in Section 5.3.1). The second baseline levers the verb-object tags

learned by the Stanford dependency parser, used as proxies for ACTION and OBJECT tags respectively. The third approach is a state-of-the-art deep semantic role labeling model with self attention [317]. Semantic role labeling is a shallow semantic parsing task that extracts various 'semantic roles', i.e. event properties and relations among relevant entities from an input utterance. In this work, we only focus on the two roles of verb and the object or entity acted upon by the verb as contributors to user intent. The second column of Table 5.2 reports the precision, recall and F1-score of the ACTION tags for each word of the input utterance, whereas the third column only assesses the OBJECT tags. The fourth column displays the results considering the combination of both tag types to create an intent. The last column of semantic similarity computes the mean of the cosine similarities between the embeddings of the predicted and actual (annotated) intents. Each intent phrase's embedding is the average of the pre-trained GloVe [246] embeddings of its constituent words. We ignore the words whose embeddings do not exist. '*beam-CRF*' and '*constr-CRF*' in the last two rows refer to the two CRF enhancements from Section 5.3.1 of (i) considering a beam of probable tag sequences, and (ii) adding additional constraints to the decoding algorithm.

We observe a significant improvement of OPINE of over 15% in terms of F1-score and semantic similarity, compared to the simple intent-indicator based model and the Stanford parser (first two rows of Table 5.2). The best performing variant of our proposed approach also significantly outperforms the SRL model (third row of Table 5.2) by about 9% F1-score points. These results show that OPINE can successfully filter out all the additional "non-intent" background information present in the input utterance, and only focus on the text needed to extract the user intent.

OPINE Design Choice Analysis: Table 5.3 describes the performance of different variations (design choices) of OPINE for open intent extraction on our curated Stack Exchange

Table 5.3: Precision (P), recall (R), F1-score and semantic similarity of OPINE variants on the Stack Exchange dataset.

OPINE design variant	**ACTION P/R/F1**	**OBJECT P/R/F1**	**Intent P/R/F1**	**Semantic Similarity**
att+adv + train on MTurk + w-dist	0.75/0.59/0.66	0.74/0.52/0.61	0.74/0.55/0.63	0.74
att + SC (pre-train) + MTurk (fine tune) + w-dist	0.78/0.62/0.69	0.79/0.56/0.66	0.78/0.58/0.67	0.80
adv + SC + MTurk + w-dist	0.81/0.60/0.68	0.76/0.54/0.63	0.78/0.56/0.65	0.77
att+adv + SC + MTurk + w-dist	0.84/0.66/0.73	0.81/0.63/0.71	0.82/0.64/0.72	0.83
att+adv+beam-CRF + SC + MTurk + w-dist	0.84/0.70/0.76	0.81/0.67/0.73	0.82/0.68/0.74	0.84
att+adv+constr-CRF + SC + MTurk + w-dist	0.84/0.72/0.77	0.81/0.67/0.73	0.82/0.69/0.75	0.85
att+adv + SC + MTurk + MLP	0.84/0.68/0.75	0.81/0.67/0.73	0.82/0.67/0.74	0.84
att+adv+beam-CRF + SC + MTurk + MLP	0.84/0.72/0.77	**0.81/0.69/0.75**	**0.82/0.70/0.76**	**0.86**
att+adv+constr-CRF + SC + MTurk + MLP	**0.84/0.73/0.78**	0.81/0.68/0.74	**0.82/0.70/0.76**	**0.86**

dataset. Except for the first row of Table 5.3, all other variants of OPINE are first pre-trained on the verb-object tags learned by a dependency parser (SC), followed by fine-tuning on the intent annotated utterances. '*train on MTurk*' denotes OPINE being trained only on the human annotated intent data. '*att*' and '*adv*' denote the presence of attention and adversarial training respectively in the model. *'w-dist'* and '*MLP*' denote the two methods of matching appropriate ACTION-OBJECT phrases to create a holistic intent, from Section 5.3.2 based on (i) word proximity in the input text, and (ii) the score learned by the MLP classifier. Utilizing the dependency parser data as a pre-training step for the weights of our model, followed by fine-tuning on the actual intent-labeled data improves the F1-score by at least 6%. Enhancing the CRF decoding algorithm with added constraints (*beam-CRF* and *constr-CRF*) benefits the F1-score further by 2-5%. We find a performance difference of $\leq 3\%$ between using the word proximity heuristic (*w-dist*), and the MLP classifier for matching ACTION and OBJECT phrases. In general, the unsupervised word proximity heuristic is more efficient than the MLP classifier because it does not incur an additional training cost. Overall, OPINE trained

with attention, adversarial training and CRF enhancements outperforms the alternative variations (Table 5.3) and state-of-the-art baselines (Table 5.2), with an intent F1 score of 76%, and a semantic similarity of 86% between the true and predicted intents.

Domain Adaptation Capability: Encountering newly emergent domains with little to no labeled intent categories available is a common real-world scenario. It is time-consuming and labor-intensive to obtain sufficient domain-specific annotated training data. It is thus desirable to adapt and generalize an existing trained model with minimum re-training effort, each time a new domain with potentially new intents is added. We investigate in Table 5.4 the capability of OPINE in adapting and transferring knowledge across distinct conversational domains. We consider several different test domains in the first column. The average overlap in the text vocabularies across pairwise domain combinations is 43%. We evaluate OPINE trained on utterances from the remaining domains other than the test domain in the second and fourth columns. The third and fifth columns indicate the respective F1-score and semantic similarity achievable for the test domain, when OPINE is trained using labeled data from the test domain as well. The F1-score and semantic similarity metrics are computed in a similar manner as for Tables 5.2 and 5.3. The difference in both metrics with and without using training data from the test domain is $\leq 5\%$, for most domains. Only the *Life Hacks* domain suffers a loss of 6.5% in terms of F1-score when we eliminate the data from this domain while training. Interestingly for the *DIY* domain, its training data is dominated by other semantically distinct domains. However, OPINE still attains a good F1-score of 72%, only 4% lesser than what is possible if *DIY* domain data is used for training. These results show that OPINE can effectively detect novel actionable intents in newly emerging low-resource domains with minimal manual effort.

Table 5.4: Studying OPINE's domain adaptation capability on multiple test domains. '+td' in the columns indicates that data from that particular test domain row is included while training, while '-td' indicates its exclusion while training.

Test Domain Name	**F1 -td**	**F1 +td**	**Sim -td**	**Sim +td**
Data science	0.76	0.8	0.84	0.88
Software engineering	0.69	0.74	0.81	0.86
Web apps	0.73	0.77	0.83	0.88
Webmasters	0.75	0.79	0.83	0.86
Sharepoint	0.71	0.76	0.82	0.85
Productivity	0.73	0.78	0.81	0.86
Development ops	0.71	0.73	0.78	0.83
Open data	0.69	0.73	0.84	0.87
Server fault	0.67	0.72	0.75	0.8
Life hacks	0.635	0.7	0.74	0.8
DIY	0.72	0.76	0.81	0.86
CRM software	0.79	0.83	0.88	0.91

Role of Attention: Table 5.3 indicates that the presence of attention lends OPINE an F1 score gain of at least 4%. We further explore OPINE's capability of identifying relevant and meaningful semantic features from its input utterances, which contribute in discovering open intents. We examine and visualize in Table 5.5 the self-attention values for specific utterances from our Stack Exchange dataset. For the sake of brevity we display truncated versions of the text inputs in the first column, and their associated intents in the second column. A darker colored highlight on a specific utterance word indicates that it receives higher attention, and plays a greater role during intent discovery. Input utterance words that constitute intents are marked in boldface. In all cases, we observe that words semantically related to and contributing to at least one intent are successfully identified by an attention head. For instance, the second row shows the significance of *'find out', 'retweeted', 'tweet'* and *'what their Twitter IDs are'* in deciding the intent *"find retweeted Twitter IDs"*. The attention heads are attentive to indicator cues that are likely to precede an actionable intent,

Table 5.5: Effect of attention. Darker colored highlight shows a higher attention value. Boldface denotes presence of intent.

Input Text Utterance	Intents
Is it possible to **navigate** back ... **to previous page** after save processing? ... I have a page where I click on a link and use navigateURL ... want to be able to go back to the previous calling page and **complete** the **processing** of the **save**...	navigate previous page, complete processing save
The "Your tweets retweeted" page ... **find** out all the users who **retweeted** a tweet of mine? ... have retweeted a tweet and what their **Twitter IDs** are?	find retweeted Twitter IDs
Is there a WordPress plugin that will **tweet when** a scheduled post is posted? ... will tweet when you **publish** a post, but none I have tried will do it on a **scheduled post**.	tweet when publish scheduled post
How can I **keep** my **phone from** just **falling** over when watching videos? ... I also want to **have** my **hands free** to do other things ...	keep phone from falling, have hands free
I'm starting a micro-school... I want to **manage sick notes** and **absences** ... How can I **synchronize** one **central** Google **Calendar** ... Parents should be able to schedule future absences and excuse past absences...	manage sick notes, manage absences, synchronize central calendar

such as *'possible to', 'want to be able to', 'how can I'* and *'I want to'*. Action or object phrases that are irrelevant to the user's intent (e.g. *'watching videos'* in the fourth row) do not receive a high attention score. Our attention mechanism can capture the dependency between distant intent words, such as *'find'* and *'retweeted'* in the second row and *'publish'* and *'scheduled'* in the third row. It also associates the action *'manage'* in the last row with two objects, *'sick notes'* and *'absences'*, generating the intents *"manage sick notes"* and *"manage absences"*.

5.4.3 Drill-down Analysis

Effect of Human-Annotated Training Data Size: In Tables 5.2 and 5.3, we showed that training our OPINE model with absolutely no human-labeled intent data is detrimental to its performance. We now examine the effect of varying the amount of human annotated

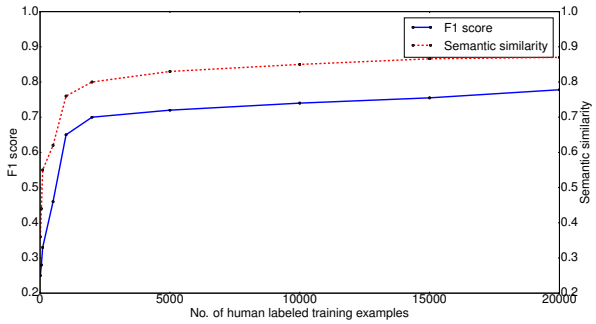


Figure 5.2: Effect of varying the amount of human labeled training data on the model performance of OPINE.

intent data while training OPINE. Figure 5.2 shows the F1-score (blue solid plot) and semantic similarity (red dashed plot) values for the predicted intents achieved by OPINE, as the number of human annotated training instances varies. We find that both plots are monotonically increasing. When the total number of human labeled training instances across various domains is less than 1000, the values of the F1-score and semantic similarity are below 50%. Both metrics rise to about 70% and 75% respectively at 1000 annotated training examples (less than 50 labeled examples per unique domain on average). Beyond this point, there is a steady performance improvement, with a less sharper gain than earlier. These observations reinforce OPINE's domain adaptivity and show that it does not require a large number of labeled examples per domain (less than 50 on average) to successfully perform intent discovery.

Grouping Related Intent Categories: The output of OPINE is an intent phrase for each input user utterance, and might therefore use distinct phrases to express similar intents. For instance, for the following semantically equivalent utterances: (i) *"Make a new haircut appointment for next Saturday"* and (ii) *"Can you reserve a time slot in the hair salon on Saturday?"*, OPINE predicts their intents as "make haircut appointment" and "reserve

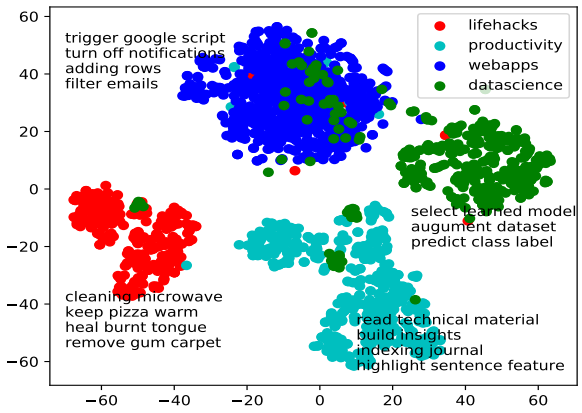


Figure 5.3: Visualizing the deep features learned by OPINE for four randomly selected Stack Exchange domains.

time hair salon". We might want to group together such semantically similar intents into a unified intent category. Therefore, as an additional post-processing step, we provide the deep features output by the attention layer of OPINE as input to an agglomerative hierarchical clustering algorithm, where the number of clusters is given by the number of distinct intent categories or domains that we would like to obtain (e.g. 12 from Table 5.1 in the case of our Stack Exchange dataset). We visualize the output vectors of the attention layer of OPINE for four random test domains from our curated Stack Exchage dataset in Figure 5.3. We use t-SNE [200] to reduce the vector dimensionality to two dimensions, and color the data points according to their domain labels. Some sample intents discovered by our method for each domain are also denoted in Figure 5.3. Domain names are treated as ground truth labels for the purpose of evaluation with respect to standard clustering metrics. We obtain a normalized mutual information (NMI) [302, 352] score of 0.71, a silhouette coefficient [269] of 0.69, a cluster purity [266] value of 0.78, and an F-measure [266] of

0.72. Figure 5.3 and the high purity value of our clustering arrangement demonstrate that good quality features are learnt by OPINE, which are largely compact within domains and separable across domains. Utterances belonging to similar intents and the same domain are also located close by in the embedding space (except for some domains like *Webapps* and *Data Science* that contain some utterances with similar intents). OPINE can thus be used to group together semantically or functionally related intents into higher level categories (the task of *domain discovery*).

5.5 Case Studies

We now present qualitative and quantitative evaluations of OPINE on additional datasets. Note that we trained OPINE on *unrelated, out-of-domain* intent categories from Stack Exchange, before testing on these datasets to examine the performance of OPINE. This represents a challenging environment, akin to zero-shot learning [236, 295], where no information about the test data is known during training.

Ubuntu Dialog Corpus: We evaluate OPINE on a real, multi-turn conversation from the Ubuntu Dialog Corpus [194], a snippet of which is shown in Table 2.4. This dataset contains about one million technical support conversations related to the Ubuntu Linux operating system, and highly resembles real-world dialog exchanges between users and customer support agents. The original dialog from which this snippet has been truncated contains more than 100 turns with a lot of extraneous background information. It is between a user with technical issues and another who helps resolve them. Such data is often asynchronous with diverse and informal intents, dialog domains and semantic slots; which increases the difficulty of intent discovery. As mentioned earlier, we used OPINE pre-trained on

Table 5.6: Performance of OPINE on a technical support dialog snippet. Words that make up intents are shown in boldface.

User: Can someone please help? I'm trying to **fix** a **broken ubuntu**.
Agent: ... how did you break it?
User: i'm on the cd and i'm trying to **mount** and then **chroot** my **hd**, which worked fine. I installed some new libs and now it no longer reboots.
User: what's the easiest way to **get** a **working boot** on my drive again?
Agent: ... sounds like something might be screwed up in your /etc/apt/sources.list file, if it's failing on apt-get update
User: how can i **fix** my **sources.list file**?
Agent: open /etc/apt/sources.list. see if you notice any obvious errors
User: a question on the mounting issue - when i loaded the cd, my local hard drive was mounted in media, can't i just use that as the chroot?
Agent: ... assuming your flgrx is hosed, move the x conf file out of the way so that the radeon driver will be used instead ...
User: ... what do you suggest for a good backup program for ubuntu?
User: ... i installed the latest radeon drivers manually. how do i **upgrade** to the **newest kernel** and default **radeon drivers**?
Agent: first you'd uninstall 10.6 fglrx driver. then you'd grab the three 2.6.34 deb packages and then install xorg-edgers repo. run grub-update so it finds the new kernel and done.
User: where do i **get** the **debs**? and i know how to **uninstall** the **fglrx drivers** ..., and then do i **copy** back the **xorg.conf.original** to xorg.conf?
User: ... do i need to add a source to my source list?
Agent: yes you need xorg-edgers (google it)
User: ok cool. how do i **get rid** of **xorg**, or is that already done?
Agent: ... if you used jockey-gtk to install fglrx and no other method, then you should be able to use the same method to remove them

unrelated intent genres from our Stack Exchange dataset, before testing on this multi-turn dialog. Our goal here is to understand the actionable intents of the user requesting support (**User** in Table 2.4), and not the one providing it (**Agent** in Table 2.4). We also seek to handle utterances containing intents with *both* an action and an object. The words constituting intents inferred by OPINE have been highlighted in boldface. Though each training utterance has up to three labeled intents, OPINE can detect more than three intents for an input if applicable. OPINE recognizes the following user intents in the conversation:

(a) PlayMusic (b) SearchCreativeWork (c) SearchScreenEvent (d) BookReservation

Figure 5.4: Fine-grained intents discovered by OPINE for four intent categories in the SNIPS NLU dataset. The length of the bars represents the relative frequency of that particular intent in the input data.

fix broken ubuntu, mount hd, chroot hd, get working boot, fix sources.list file, upgrade newest kernel, upgrade radeon drivers, get debs, uninstall fglrx drivers, copy xorg.conf original and *get rid xorg*. Once these fine-grained intents have been recognized, they can be subsequently grouped into coarser level intents or domains (for instance, using the technique for Figure 5.3), depending on the downstream application task. Categorizing such a multi-turn dialog is typically outside the scope of existing intent classification systems, but OPINE provides a fine-grained summary of user intents throughout the dialog. Besides, having a common format to represent an intent contributes immensely in finding user intents irrespective of their target domain or topic.

Performance on SNIPS and ATIS: We next discuss the performance of OPINE on standard intent detection datasets used in the literature, namely the SNIPS NLU [61] and ATIS [62] datasets. We highlight in Figure 5.4 the benefit of OPINE in drilling down into high-level intent categories, to understand, summarize or hierarchically organize the specific fine-grained intents that they comprise of. An additional side benefit of discovering intents using OPINE is that it can identify relevant accompanying *slots* apart from the intents, without performing a dedicated slot filling task. For instance, in the *PlayMusic* category of the SNIPS dataset in Figure 5.4(a), OPINE not only recognizes the basic intents of *'hear song'*

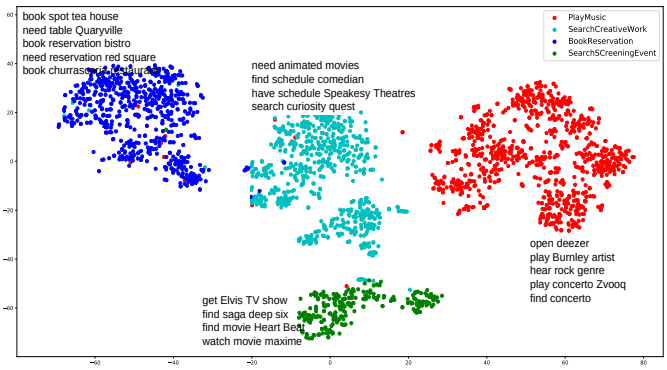


Figure 5.5: Visualizing the clustering arrangement for utterances belonging to four intent categories in the SNIPS NLU dataset. This illustrates OPINE's ability to conflate different intent phrases mapping to the same or similar intent category. E.g. 'book spot tea house', 'need table', 'need reservation' and 'book reservation bistro' map to the same cluster.

or *'play album'*; but also the corresponding names of singers (e.g. *Leroi Moore, Eddie Vinson*), song albums (e.g. *Curtain Call*, *Concerto*), and music platforms (e.g. *Youtube*, *Zvooq*). In Figure 5.4(c), users wish to search for screenings of events. OPINE accurately predicts this via keywords like *search, locate, find* and *look*. Moreover, OPINE also provides added information on the specific events that need to be searched, such as the *Chump Change* saga and the movie *Heart Beat*.

We emphasize here that OPINE *cannot* be directly compared to existing intent detection techniques in general since these are formulated as classification tasks over a fixed set of pre-defined intent categories, that need to be known at training time. They also typically require sufficient labeled training data for each intent category. In contrast, OPINE can handle an unlimited number of distinct intent classes, and has no restrictions on the number of training examples needed per intent type. However, to compare OPINE with existing intent classification approaches, we reformulate it to perform intent detection, using the output of the multi-head attention layer (OPINE + classifier in Table 5.7). As detailed

Table 5.7: F1 score of various intent classification approaches on the SNIPS and ATIS datasets.

Approach	**SNIPS**	**ATIS**
Joint Seq [330]	0.95	0.91
Attention BiRNN [186]	0.95	0.90
Slot-Gated Full Atten. [101]	0.95	0.92
DR-AGG [100]	0.95	0.90
IntentCapsNet [369]	**0.96**	**0.94**
OPINE + classifier	**0.96**	0.93
OPINE + Joint-Seq	0.95	0.91
OPINE + Attention Bi-RNN	0.95	0.90
OPINE + Slot-Gated Full Atten.	0.95	0.92
OPINE + IntentCapsNet	0.95	0.93

earlier, we provide the deep features output by the attention layer of OPINE as input to an agglomerative hierarchical clustering algorithm, where the number of clusters is given by the number of distinct intent classes in the test set. Thus "OPINE + classifier" works as a good feature extraction layer across datasets without additional training, which is a benefit over current state-of-art techniques.

Note that OPINE was trained on *out-of-domain* intent data (Stack Exchange), since we do not have ACTION-OBJECT annotations available for SNIPS or ATIS. In other words, this represents a challenging environment to examine the performance of OPINE (akin to zero-shot learning [236, 295]), where no information is available about the test data. Similar to what we showed in Figure 5.3 for the Stack Exchange dataset, in Figure 5.5 we show OPINE's ability to conflate different intent phrases mapping to the same or similar intent category. We show a 2-D t-SNE visualization of the deep features output by the attention layer of OPINE for the SNIPS dataset. For instance, the intent phrases *play Burnley artist* and *hear rock genre* map to the same intent category (called *PlayMusic* in the SNIPS dataset),

while the phrases *book reservation bistro* and *need table Quaryville* map to the same SNIPS category of *BookRestaurant*.

In addition to this qualitative analysis, we present a quantitative evaluation on SNIPS and ATIS in Table 5.7. It shows that the F1-score of OPINE is on par with state-of-the-art intent classification approaches on these datasets, despite not having seen any SNIPS or ATIS data during training. The SNIPS and ATIS utterance lengths are also about 2-6 times shorter than the Stack Exchange texts. This shows that OPINE is equally effective at finding intents in shorter inputs with no or limited additional context. The last four rows in Table 5.7 show the performance of the baselines when the intent-tagged words from the output of OPINE are fed to them as inputs. These inputs only contain the words tagged as ACTION or OBJECT by OPINE, without any additional context. Their lengths range from 20-50% of the original input utterance. We find that just using OPINE as an initial step before intent classification has near identical performance as the baselines themselves (rows 1, 2, 3 and 5 of Table 5.7), despite OPINE not having seen any SNIPS or ATIS data during training. This further demonstrates the domain adaptation capability and effectiveness of OPINE.

5.6 Related Work

We discuss prior work on intent detection, as well as the related tasks of information extraction and semantic parsing.

Intent Detection: Prior work on intent detection encompasses two avenues: asynchronous, written communication (forums, blogs, tweets) and synchronous dialog. In both cases, intent detection is largely modeled as a classification problem, with each class representing the presence or absence of a specific kind of intent. Supervised and semi-supervised learning models based on linguistic and sentiment features have been used to model racial

intent [3], and purchase intent online [54,112,358]. Wang et al [358] proposed a graph-based optimization approach for semi-supervised classification of tweets with a purchasing intent. Recent approaches including automated dialog response agents like Microsoft LUIS[11] and Google Dialogflow[12] have drawn upon progress in CNN, RNN, and transformer based language models to improve intent detection [41, 101, 156, 158, 186, 212, 288, 372] and domain detection [155, 157, 394]. Performing slot filling jointly with intent detection has improved the performance of both tasks [81, 156, 186, 359, 398]. However, the above approaches are either supervised or semi-supervised, assume a closed-world setting, and require sufficient quantities of labeled data for each intent type to perform intent detection.

In the open world setting, several interesting techniques were proposed to recognize input texts belonging to unseen or novel domains [155] and intents [181, 369] respectively. Xia et al [369] model intent detection as a zero-shot classification problem, but they assume that a list of new or unseen (during training) intent classes is available at test time along with some prior knowledge about the new intents to be discovered. Similarly, the methods proposed by Lin et al [181] and Kim et al [155] can only identify if an input utterance is likely to contain a novel intent. They do not discover the number of new intents, or specify what the new intents are. OPINE takes the next step in this regard.

Cai et al [37] used hierarchical clustering to learn a taxonomy of intent classes [145, 338, 340], and applied a hybrid CNN-LSTM model to classify the intent of medical queries. We take this idea further and learn to identify arbitrary intents beyond even a predefined taxonomy or schema. Improvements in intent detection and slot filling based on adversarial learning have been explored [159, 187, 388]. We exploit adversarial training to generate

[11] `www.luis.ai`

[12] https://dialogflow.com/

adversarial input examples to improve the performance of OPINE, and for cross-domain adaptation. Further, adding linguistic structure to existing models has improved their performance across related NLU tasks such as word embedding [220], machine translation [49], named entity recognition [140], and semantic role labelling [119]. We impose linguistic constraints on the CRF layer of OPINE to preserve the semantics of intent actions and their associated objects (Section 5.3.1).

Open Information Extraction and Semantic Parsing: There has been a lot of recent work in the literature in the areas of Open Information Extraction (IE) (see [231] for a recent survey). IE addresses the task of information extraction, i.e., of relating arguments and phrases expressed in unstructured text using a relation of the form <*arg1, rel, arg2*>. OpenIE refers to the task of building domain independent IE frameworks, where the relations to be extracted need not be specified in advance. In OpenIE frameworks, relations are expressed as triples and each component of the triple must be present in the input text. However, this is quite different from the open intent discovery task we define in this work in the following ways: First, an OpenIE model does not distinguish between pieces of information that express an *intent*, and those that do not. Post-processing of relevant relations would be necessary to identify expressed intent before OpenIE systems could be used for the intent discovery task. Second, existing OpenIE systems extract information only at the level of sentences, unlike our proposed approach which can extract intents spanning *multiple* sentences (see Table 5.5 for examples). For these reasons, we do not compare our proposed approach OPINE with existing OpenIE models.

Semantic Role Labeling (SRL) (and also slot filling) approaches aim at creating shallow semantic parses, assigning roles to different phrases in input sentences. Semantic roles typically correspond to slots in predefined frames/templates (eg. Propbank [26]). Recent

work on SRL aims to relax the assumption of pre-specified templates to move towards out-of-domain SRL [116, 119, 303]. Though SRL is a task related to intent discovery, SRL labels are typically more dense, requiring not just the intent labels, but other subject-predicate, named entity, etc. relations to be labeled. However for the purpose of completeness in evaluation, we do compare our framework against a state-of-the-art SRL baseline [317] in Table 5.2. OPINE siginificantly outperforms this baseline for intent discovery.

5.7 Conclusion

We introduce and address the novel problem of open intent discovery via a sequence tagging approach, OPINE, in contrast to the common method of detecting intents via classification. OPINE harnesses a Bi-LSTM and a CRF coupled with self-attention and adversarial training. It can extract multiple actionable intent types from user utterances, many of which may be unseen during training. Extensive experiments on real-world datasets show substantial improvements of OPINE over competitive baselines. We also developed and plan to release a large collection of 25K intent-annotated instances from diverse domains on Stack Exchange. We demonstrate OPINE's ability to adapt across domains, minimizing the labeling effort needed on encountering a new domain with potentially new intents. OPINE provides an in-depth, fine-grained understanding of users' prospective actions and intentions from their text utterances, which can greatly benefit downstream conversational applications. Promising future directions include (i) learning generative rather than extractive models for open intents; (ii) inferring implicitly mentioned, or more generic non-actionable (e.g. information-seeking) intents from user utterances; and (iii) using visual and/or auditory inputs to learn intents in a multimodal way (e.g. [346, 362]).

Chapter 6: ADVIN: Automated Discovery of Novel Domains and Intents in User Text Utterances

Recognizing the intents and domains of users' spoken and written language is a key component of Natural Language Understanding (NLU). Most existing research formulates this as a supervised classification problem with a *closed-world* assumption, i.e. the domains or intents to be identified are pre-defined beforehand. Real applications however increasingly encounter dynamic, rapidly evolving environments with newly emerging intents and domains, for which labeled data is not available. In the previous Chapter 5, we proposed a domain-agnostic technique OPINE, to automatically discover generic actionable intents within user text utterances, irrespective of the amount of labeled training data available for them. However, OPINE had the following limitations: (i) it is restricted to intents containing an action or activity to be performed; and (ii) does not categorize the novel intents discovered into novel domains. In the current Chapter, we proposed a novel framework, ADVIN, to tackle these limitations. It automatically discovers novel domains and intents from large volumes of unlabeled text. We first employed an open classification model to discriminate all utterances potentially consisting of a novel intent. Next, we trained a deep learning model with a pairwise margin loss function and knowledge transfer, to discover multiple *latent* intent categories in an unsupervised manner. We finally formed a hierarchical taxonomy by linking mutually related novel intents into novel domains. ADVIN significantly outperforms

competitive baselines on four benchmark datasets, and real data from a commercial voice-powered agent.

6.1 Introduction

Numerous everyday gadgets like mobile phones or smart speaker devices consist of an NLU component to automatically understand and service user requests. . Comprehending the *intent* and/or *domain* (groups of mutually related intents) of users' language utterances is a key task in these devices. Various techniques have been proposed in the literature for this. But most methods are supervised or semi-supervised, i.e. they capitalize on sufficient labeled data, can only handle a fixed number of intents and domains *seen* during model training, and generalize poorly to new intents or domains *unseen* during model development. However, continual user interactions with voice-powered agents often contain new domains or intents not encountered by the learning models before. Expanding the capabilities of the above models would require expensive human labeling efforts each time a new domain or intent is encountered.

Zero shot techniques [167, 369] recognize new intents for which no labeled training data is available. However, they require some (often unfeasible) additional information like the number of new intent types, and some prior knowledge about the new intents to be discovered. Efforts have been made to break the closed-world assumption in the NLU literature [155, 181, 290, 336]. This is the paradigm of *open world learning* or *open classification*, that identifies instances with labels unseen during training. However, the task of discovering the actual latent categories within the instances identified as possessing unseen labels is relatively under explored. Shu et al [291] attempted this for image classification, but their model could not always outperform baselines. In this work, we attempt to bridge the gap

between the two challenging yet realistic tasks of (i) discriminating utterances belonging to new intents/domains from utterances belonging to already familiar ones, and (ii) organizing the newly discovered intents/domains into a taxonomy. Though we address the problem of novel user intent and domain discovery, our technique can easily be generalized to any open classification setting.

We propose a novel, three-step framework called *ADVIN* (Automated Discovery of noVel domaIns and iNtents). It automatically discovers user intents and domains in massive, unlabeled text corpora, *without* any prior knowledge about the intents or domains that the text may comprise of. Our method first leverages the pre-trained multi-layer transformer network, BERT [69], to determine if an utterance is likely to contain a novel intent or not. ADVIN next uses unsupervised knowledge transfer to discover the latent intent categories in the earlier identified utterances. Finally, ADVIN hierarchically links semantically related groups of newly discovered intents to form new domains. To summarize:

(i) We propose a novel, fully automated method, ADVIN, that correctly identifies texts containing novel intents and novel domains, *without* any labeled training data for the novel intents and domains.

(ii) To the best of our knowledge, ADVIN is the first work to generalize to diverse, low-resource, open-world scenarios, and is independent of both the intents and domains that it has been trained upon.

(iii) We extensively evaluate ADVIN on public benchmark datasets and real-world data from a commercial voice agent, and significantly outperform baselines across various empirical configurations.

6.2 Related Work

Intent detection has been successfully performed in the literature via numerous machine learning approaches [41, 137, 156, 181, 258, 288, 309, 331, 336, 369]. Intent detection has also been jointly done with slot filling to improve its performance [21, 101, 158, 186, 359, 398]. However, the above approaches either require sufficient labeled data for each domain and intent (supervision), or some prior knowledge about the new intents to be discovered (zero shot). Most prior work cannot jointly detect novel intents and domains. ADVIN seeks to eliminate these restrictions.

Deep clustering networks [128] and Bayesian non-parametric models [4, 73] were proposed to identify mixture components or clusters in data. But unlike ADVIN, the clusters identified cannot automatically be mapped to unique classes. Our work is also different from semi-supervised clustering techniques [88, 128, 385], which lever some labeled data and require the classes to be discovered to be known beforehand.

6.3 Our Framework ADVIN

Formally, we are given a corpus of utterances $\mathscr{D}_{\mathscr{T}}$ labeled with S seen intents and S_D seen domains; and a corpus of unlabeled utterances $\mathscr{D}_{\mathscr{C}}$ consisting of U novel intents and U_D novel domains (Figure 6.1). $S \cap U = \varnothing$ and $S_D \cap U_D = \varnothing$. We propose a three-stage framework ADVIN that (i) classifies the incoming utterance $x \in \mathscr{D}_{\mathscr{C}}$ with one of the

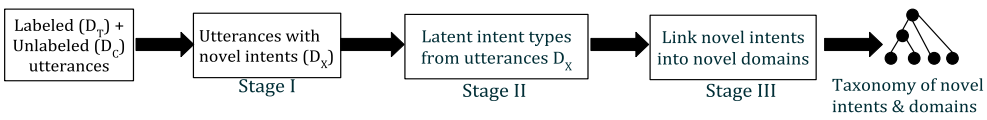


Figure 6.1: Overall pipeline of our proposed approach, ADVIN

$S = (s_1, ..., s_S)$ seen intent labels or as an *unseen* (*novel*) intent; (ii) for utterances $\mathscr{D}_{\mathscr{X}}$ predicted as having a novel intent, it discovers the latent intent groups $U = (u_1, ..., u_U)$ present in them; and (iii) links related novel intents discovered to form novel domains.

6.3.1 Stage I: Detecting Instances Containing Novel Intents

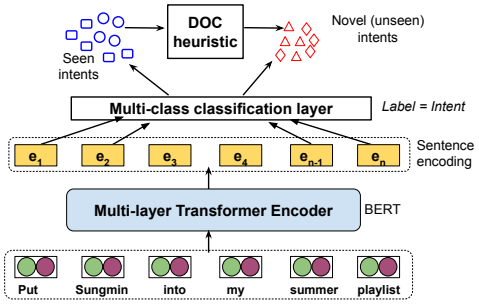
We construct a two-step system to detect the presence of *novel* intents U in input instances, *without* any labeled training examples for these novel intents. This is a challenging yet realistic problem setup, since a learned model is likely to be biased towards the seen intent classes S for which labeled data exists.

Step I: Prior open classification literature has modeled the problem of finding *novel* or *unseen* classes as an $(S+1)$-class classification problem, with S *seen* classes and an additional *unseen* class. Labeled training data is available for the seen classes, and an input instance is classed as unseen if it does not belong to any seen class [290, 291, 371]. Learning a good distribution of the labeled data belonging to the seen intents is necessary to distinguish the unlabeled texts containing novel intents. We therefore effectively leverage vast amounts of linguistic and contextual knowledge from the pre-trained language model BERT [69], to learn an $(S+1)$-class classifier (Figure 6.2a), that classifies utterances as containing a novel intent or not. An input utterance $\boldsymbol{x}$ is converted into a sequence of tokens to be input to BERT, with a special start-of-sequence token, "`[CLS]`", as the first token. The output of BERT is an n-dimensional utterance encoding $(\boldsymbol{e_1}, ..., \boldsymbol{e_n})$. A multi-class softmax classification layer predicts the utterance intent based on the final hidden state $\boldsymbol{e_1}$ of the `[CLS]` token. We fine tune the parameters θ in different BERT layers with different learning rates as per [126]: $\theta_t^l = \theta_{t-1}^l - \eta^l \nabla_{\theta_l} J(\theta)$. Here η_l and θ_l denote the learning rate and parameters respectively

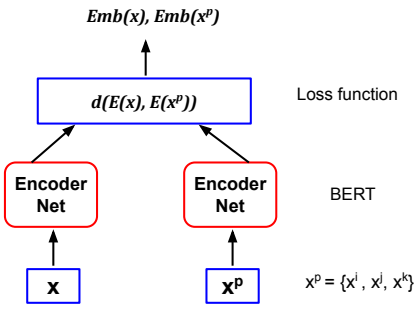
of the l-th BERT layer; and $\eta^{l-1} = \xi.\eta^{l}$. ξ is a decay factor ≤ 1, while $\nabla_{\theta_l} J(\theta)$ is the gradient with respect to the model's objective function.

For our $(S+1)$-class classifier described above, labeled training data $\mathscr{D}_{\mathscr{T}}$ is available for the S seen intents. We use *out-of-domain* (OOD) intent detection datasets distinct from both $\mathscr{D}_{\mathscr{T}}$ and $\mathscr{D}_{\mathscr{C}}$, as training data for the $(S+1)$-th 'novel' intent class. This OOD data comes from out-of-domain, intent-labeled publicly available datasets (e.g. SNIPS [61], ATIS [62]) that do not require extra human annotation effort. Note that while training the $(S+1)$-class classifier, ADVIN only requires the information that the OOD intents do not overlap with those in $\mathscr{D}_{\mathscr{T}}$, and *not* the actual intent labels of the OOD data. However, if the OOD data (e.g. SNIPS) is annotated with m intent classes, ADVIN can use this information by fragmenting the $(S+1)^{\text{th}}$ class into m classes, forming an $(S+m)$-class classifier. Using m classes may provide a better representation of the texts containing unseen intents (see Table 6.2). An utterance classified into any of the m classes is flagged as having a novel intent.

Step II: To learn an effective distribution of the data belonging to the seen and novel intents, we adopt an additional step inspired by the Deep Open Classification algorithm (DOC) [290]. After Step I classifies the input as having a seen or novel intent, ADVIN learns statistical confidence thresholds for each seen intent s_i. It thus captures instances that have been classified to one the S seen intents with a low confidence. If the class-specific prediction probabilities for an utterance are less than the thresholds learned for each seen intent s_i, that utterance is also classified as having a novel intent (Figure 6.2a).



(a) Detecting instances containing novel intents



(b) Discovering novel intent categories

Figure 6.2: Overview of both stages I (a) and II (b) of our proposed approach ADVIN

6.3.2 Stage II: Discovering the Latent Novel Intent Categories from the Unlabeled Instances

Stage I gives all utterances $\mathscr{D}_{\mathscr{X}}$ that potentially contain a novel intent. Next, we discover the actual latent intent categories U within $\mathscr{D}_{\mathscr{X}}$. We use complete-linkage, agglomerative hierarchical clustering [103] to group together related utterances in $\mathscr{D}_{\mathscr{X}}$, and find the potential novel intents U.

Knowledge Transfer Component: Intuitively, humans seem to understand and categorize newly encountered objects based on the characteristics that they have learnt from their prior knowledge of similar or comparable objects [63, 110]. We utilize a similar idea in order to learn an effective distance threshold δ at which to stop the agglomerative hierarchical clustering. That is, we transfer the knowledge learned from clustering the utterances containing seen intents S, to the utterances containing novel intents. We assume that utterances in the training set $\mathscr{D}_{\mathscr{T}}$ of seen intents, as well as those in the unlabeled corpus $\mathscr{D}_{\mathscr{X}}$ of newly emerging unseen intents come from similar distributions. First, we hierarchically cluster the labeled training data utterances $\mathscr{D}_{\mathscr{T}}$, using the seen intents as

ground truth cluster labels. We obtain a distance value δ by maximizing the F1 score of their clustering arrangement. This maximizes the distance between utterance clusters containing distinct intents, and minimizes the distance between utterances containing the same intent within a cluster. In an ideal scenario, every obtained cluster L_i represents a single seen intent. Our final step is then to *transfer* this distance threshold δ learnt from the *seen* intent utterances, to hierarchically cluster utterances in $\mathscr{D}_{\mathscr{X}}$ containing *novel* or unseen intents. This distance threshold δ is defined as $\max_{x \in L_i, x^p \in L_j} f(Emb(\boldsymbol{x}), Emb(\boldsymbol{x^p}))$ according to complete-linkage hierarchical clustering. Here, function $f(.)$ quantifies the distance between the embeddings $Emb(.)$ of an utterance pair $(\boldsymbol{x}, \boldsymbol{x^p})$ belonging to clusters L_i and L_j respectively.

We now describe how we learn the distance function f between pairs of utterances.

Learning Pairwise Utterance Distances for Clustering: We learn a neural network model to obtain utterance embeddings $Emb(.)$ for both the labeled corpus of seen intents ($\mathscr{D}_{\mathscr{T}}$) and the unlabeled utterances detected to have novel intents ($\mathscr{D}_{\mathscr{X}}$). We utilize the knowledge from both the seen intents S as well as seen domains S_D to train this model (see Figure 6.2b). We create a training dataset from $\mathscr{D}_{\mathscr{T}}$ comprising pairs of utterances (x, x^p). Both x and x^p contain a seen intent. For each x, there are three possible choices for its paired utterance x^p: (i) x^i containing the same domain and intent as x, (ii) x^j containing the same domain but different intent than x, and (iii) x^k containing a different domain and intent than x. Thus, $x^p \in \{x^i, x^j, x^k\}$. These utterance pairs (x, x^p) are fed as input to an `EncoderNet`, which consists of a BERT transformer block. We use the same learning rate decay strategy of fine-tuning the BERT layers as Stage I. Representations $E(\boldsymbol{x})$ and $E(\boldsymbol{x^p})$ are learned by the second last BERT layer (Figure 6.2b). The next layer on top of the `EncoderNet` blocks uses a distance function d to compute pairwise representation distances,

subject to the following bi-directional constraints: (i) the distance $d(E(x), E(x^i))$ between the representations of x and x^i should be less than $d(E(x), E(x^j))$; (ii) $d(E(x), E(x^i))$ should be less than $d(E(x), E(x^k))$; and (iii) the distance $d(E(x), E(x^j))$ between the representations of x and x^j should be less than the distance $d(E(x), E(x^k))$ between the representations of x and x^k. These constraints use linguistic and semantic relationships to ensure that utterances containing distinct domains or intents should be more distant in the learned embedding space from utterances containing the same domains or intents. We then formulate a loss function $\mathfrak{L}$ to train our model:

$$\mathfrak{L} = \frac{1}{M} \sum_{i,j,k} \{\max[0, m_1 + d(E(x), E(x^i)) - d(E(x), E(x^j)] + \alpha \max[0, m_2 + d(E(x), E(x^i)) - d(E(x), E(x^k)] + \beta \max[0, m_3 + d(E(x), E(x^j)) - d(E(x), E(x^k)]\}$$

where m_1, m_2, m_3 are predefined margins, α and β are predefined weighting scalars and M is the total number of utterance pairs (x, x^p). We found such a loss formulation to outperform the popular contrastive loss [113] and triplet loss [280] functions (see Tables 6.3 and 6.4). Next, a non-linear activation followed by a linear layer outputs embeddings $Emb(\boldsymbol{x})$ and $Emb(\boldsymbol{x^p})$ for the pair (x, x^p). Finally, pairwise distances $f(Emb(\boldsymbol{x}), Emb(\boldsymbol{x_p}))$ are computed between all utterance pairs in this embedding space, to be given as input to hierarchical clustering. Note that we do not require any intent or domain labels for the unlabeled corpus $\mathscr{D}_{\mathscr{X}}$ containing the unseen (novel) intents.

6.3.3 Stage III: Linking Mutually Related Novel Intents into Novel Domains to form a Taxonomy

We finally aim to link together the novel intents discovered in Stage II, sharing the same broad functionality or semantics, into *domains*. Hierarchical clustering can already 'merge' together intents at the upper levels of the hierarchy based on distance. However, we find that

using the domain labels available for the seen intents more directly, better categorizes the related novel intents into novel domains.

(i) Assuming an ideal clustering in Stage II (Section 6.3.2), each seen intent-cluster L_i will contain utterances belonging to a single seen intent $s_i \in S$. The domain label of cluster L_i would be the seen domain $\in S_D$ of intent s_i itself. However, L_i may not be completely pure, i.e. it may contain utterances with varied seen intent labels. In such cases, we assign the domain label for the seen intent-cluster L_i as the domain of the intent of the majority of the utterances in L_i.

(ii) We next obtain a representation $Emb_L(L_i)$ for each seen intent cluster and each novel intent cluster, L_i, as the average of the embeddings $Emb(\boldsymbol{x})$ (from Figure 6.2b) of all utterances $x \in L_i$.

(iii) Finally, we re-use our Knowledge Transfer component to cluster the representations $Emb_L(L_i)$ of the seen intent-clusters L_i themselves. This time we use the seen *domains* as ground truth cluster labels (instead of the seen *intents* as in Section 6.3.2). As earlier, we obtain a distance threshold δ that maximizes the F1 score for the seen domains. We transfer this threshold to perform hierarchical clustering of the novel intent-clusters. Each cluster so obtained contains groups of related novel intents, representing novel domains. Thus, ADVIN creates a taxonomy of novel intents and domains from unlabeled utterances.

6.4 Evaluation

6.4.1 Datasets and Experimental Setup

We test ADVIN on the real-world intent and domain detection datasets of SNIPS [61], ATIS [62], Facebook's task-oriented parsing (FTOP) data [111], NLU evaluation data (NLUED) [189] and Internal NLU Data[13]from a commercial voice assistant. For evaluation

on SNIPS, ATIS, NLUED and the Internal data, we completely remove all utterances associated with certain random sets of intents and domains from the training and validation sets of ADVIN (Table 6.1). The FTOP dataset has some utterances labeled as 'unsupported', so we simply remove these while training ADVIN. Treating these removed intent and domain types as *novel*, we assess ADVIN in discovering these intents and domains during the testing phase. As seen in Table 6.1, the novel ATIS intents are relatively semantically similar to each other, while the other datasets contain dissimilar intent types. Our experiments thus holistically exhibit the performance of ADVIN when the novel intents being discovered have varying degrees of similarity with each other (or with the existing 'seen' intents).

Hyperparameters: We now outline the exact implementation details of our algorithms, which were set based on ADVIN's performance on validation datasets. We used the English uncased BERT-Base model in all stages of ADVIN. It has 12 transformer layers, 768 hidden states, and 12 self-attention heads. We kept the dropout probability at 0.1 and used the Adam optimizer [160] with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We used slanted triangular learning rates [126] for BERT with the base learning rate at $2e^{-5}$, and warm-up proportion at 0.1. We empirically set batch sizes to 64, and the number of training epochs to 8 for Stage I, and 15 for Stage II. To learn pairwise distances between utterances for clustering in Stage II, in the loss function $\mathfrak{L}$, we set $\alpha = \beta = 1$, and $m_1 = m_2 = m_3 = 0.05$. In Stage II, we used cosine similarity as the function '*d(.)*' while computing $d(E(\boldsymbol{x}), E(\boldsymbol{x^p}))$, and euclidean distance as the function '*f(.)*' while computing $f(Emb(\boldsymbol{x}), Emb(\boldsymbol{x^p}))$.

[13]User text utterances from a voice-powered virtual assistant, whose name we omit for the double-blind review purpose.

Table 6.1: Evaluating ADVIN on discovering novel intents and domains removed during training.

Dataset	Sets of domains removed from training data for evaluation. The intents within these domains are also removed from ADVIN's training data.	# of data samples	Vocab size	Avg. text len.
SNIPS	**Set 1**: *Weather, Restaurant*; **Set 2**: *AddToPlaylist, RateBook*	13.8K	10.9K	9.05
ATIS	**Set 1**: *airline, meal, airfare, day-name, distance*; **Set 2**: *flight-time, flight-no, flight, aircraft, ground-service*	5.87K	0.87K	11.2
FTOP	**Set 1**: *unsupported, unsupported-event, unsupported-navigation, unintelligible*	44.78K	16.7K	8.93
NLUED	**Set 1**: *Email, Cooking, Transport*; **Set 2**: *Alarm, Audio, Calendar*	11.1K	9K	6.84
Internal NLU Data	**Set 1**: *Weather, Calendar, Todos*; **Set 2**: *Bookings, Sports, Search, Video, Media*; **Set 3**: *Recipe, Music, Shopping, Communication*; **Set 4**: *Global, Knowledge*	3.16M	26.7K	3.72

6.4.2 Baselines and Evaluation Metrics

We compare Stage I (Section 6.3.1) of ADVIN of detecting the presence of novel intents in input utterances, with existing approaches, as well as variants of ADVIN in Table 6.2, using the standard F1-score metric:

(i) **DOC** [290]: uses a CNN with a one-vs-rest sigmoid layer on top. It tightens the sigmoid decision boundary by learning class-specific confidence thresholds to detect novel intents.

(ii) **IntentCaps** [369]: capsule neural networks in a zero shot setting to discover novel intents.

(iii) **LOF-CL** [181]: uses local outlier detection on top of a Bi-LSTM trained with a large margin cosine loss function to classify intents seen and unseen during training.

(iv) **ADVIN *(1-unseen)*/ADVIN *(1-unseen + DOC)***: variants of ADVIN using an *(S+1)*-class classifier in Stage I (Section 6.3.1), with and without the additional check per the DOC heuristic (Step II of Stage I).

Table 6.2: F1-score of various approaches for detecting if an utterance contains a novel intent (Stage I).

Approach	**SNIPS**		**ATIS**		**FTOP**	**NLUED**	**Internal Dataset**			
	Set1	**Set2**	**Set1**	**Set2**	**Set1**	**Set1**	**Set1**	**Set2**	**Set3**	**Set4**
DOC [290]	0.73	0.69	0.71	0.7	0.76	0.73	0.7	0.73	0.72	0.71
IntentCaps [369]	0.81	0.77	0.7	0.75	0.8	0.81	0.82	0.83	0.78	0.8
LOF-CL [181]	0.79	0.73	0.68	0.74	0.78	0.8	0.84	0.8	0.82	0.81
ADVIN *(1-unseen)*	0.76	0.73	0.68	0.72	0.78	0.76	0.77	0.75	0.79	0.8
ADVIN*(1-unseen+DOC)*	0.85	0.81	0.73	0.8	0.87	0.84	0.86	0.87	0.88	0.86
ADVIN *(m-unseen)*	0.78	0.75	0.7	0.75	0.8	0.85	0.8	0.8	0.82	0.84
ADVIN *(as proposed)*	**0.9**	**0.87**	**0.78**	**0.84**	**0.9**	**0.89**	**0.9**	**0.92**	**0.9**	**0.9**

Table 6.3: Discovering the latent intent types for utterances with novel intents (Stage II). '#int.' shows the number of discovered intents, 'GT' denotes the true number of intents, and 'Pur.' denotes cluster purity.

Approach	**SNIPS Set 1 (GT = 2)**				**SNIPS Set 2 (GT = 2)**				**FTOP Set 1 (GT = 4)**			
	#int.	**NMI**	**Pur.**	**F1**	**#int.**	**NMI**	**Pur.**	**F1**	**#int.**	**NMI**	**Pur.**	**F1**
ADVIN *(clf+hier)*	**3**	0.78	0.9	0.76	**3**	0.7	0.8	0.69	24	0.4	0.56	0.38
ADVIN*(triplet+hier)*	4	0.71	0.81	0.69	5	0.65	0.76	0.66	48	0.36	0.51	0.35
ADVIN *(ProdLDA)*	NA	0.71	0.84	0.72	NA	0.66	0.79	0.68	NA	0.42	0.53	0.38
ADVIN *(proposed)*	**3**	**0.8**	**0.92**	**0.78**	**3**	**0.72**	**0.83**	**0.71**	**19**	**0.46**	**0.61**	**0.51**

(v) **ADVIN *(m-unseen)***: using an *(S+m)*-class classifier, without the DOC heuristic (Step II) in Stage I.

We do not compare ADVIN with the novel intent detection method of [336] since it only detects intents containing explicit *actions*, as mentioned in Section 6.1, which makes up $<$70% of our data.

We next evaluate Stages II and III of discovering the actual intent categories in user utterances identified by Stage I, and linking the newly discovered intents into domains. As

Table 6.4: Discovering the actual, latent novel intents and novel domains of input utterances (Stages II and III). For both datasets, the first two columns show the number of new intents (#int.) and new domains (#dom.) discovered respectively. 'GT (d, i)' denotes the true number of domains d and intents i.

Approach	**Internal Data Set 1 (GT = 3, 22)**					**NLUED Set 1 (GT = 3, 10)**				
	# int.	**#dom.**	**NMI**	**Purity**	**F1**	**# int.**	**#dom.**	**NMI**	**Purity**	**F1**
ADVIN *(clf+hier)*	108	35	0.53	0.69	0.48	86	24	0.59	0.7	0.56
ADVIN *(triplet+hier)*	206	51	0.52	0.63	0.41	134	39	0.54	0.68	0.49
ADVIN *(ProdLDA)*	NA	NA	0.56	0.7	0.55	NA	NA	0.58	0.73	0.58
ADVIN *(as proposed)*	**74**	**29**	**0.6**	**0.75**	**0.6**	**45**	**16**	**0.63**	**0.8**	**0.64**

Table 6.5: Replacing hierarchical clustering (Stage II of ADVIN) with other techniques. We show number of novel intents discovered (I), cluster purity (P) and F1 score. S1 and S2 denote Sets 1 and 2 respectively.

Clustering method in Stage II(Sec 6.3.2)	**SNIPS S1 (I/P/F1)**	**ATIS S1 (I/P/F1)**	**FTOP (I/P/F1)**	**NLUED S1 (I/P/F1)**	**Internal Dataset (I/P/F1)**	
					Set1	**Set2**
X-means clust.	9/0.84/0.7	22/0.6/0.6	54/0.5/0.4	88/0.64/0.52	133/0.6/0.5	234/0.66/0.47
HDBSCAN	21/0.8/0.67	43/0.5/0.5	91/0.5/0.4	111/0.5/0.48	165/0.5/0.5	261/0.58/0.43
RCC	**3**/0.9/0.77	15/**0.7/0.7**	30/**0.6**/0.4	56/0.71/0.56	91/0.73/0.5	188/0.74/0.5
ADVIN	**3/0.92/0.78**	**13/0.7/0.7**	**19/0.6/0.5**	**45/0.8/0.64**	**74/0.75/0.6**	**167/0.75/0.55**

per our knowledge, work in the literature only detects if utterances contain novel intents or not, and does not identify groups of latent intent categories in unlabeled text. Thus, we compare ADVIN with its own variants in Tables 6.3 and 6.4:

(i) **ADVIN *(clf+hier)***: uses the representation learned by the 2nd to last BERT layer of our Stage I classification model in Figure 6.2a, as input to hierarchical clustering in Stages II and III.

(ii) **ADVIN *(triplet+hier)***: uses embeddings learned by a triplet network [280] as input to hierarchical clustering. The inputs to the network are utterance triplets (x, x^-, x^+). x^- and x^+ contain the same domain and intent, and different domain and intent as utterance x respectively.

(iii) **ADVIN *(ProdLDA)***: uses a neural topic modeling method ProdLDA [297] to discover novel intents, instead of clustering. ProdLDA needs the number of topics as input, so we give it the number of clusters output by "ADVIN *(as proposed)*".

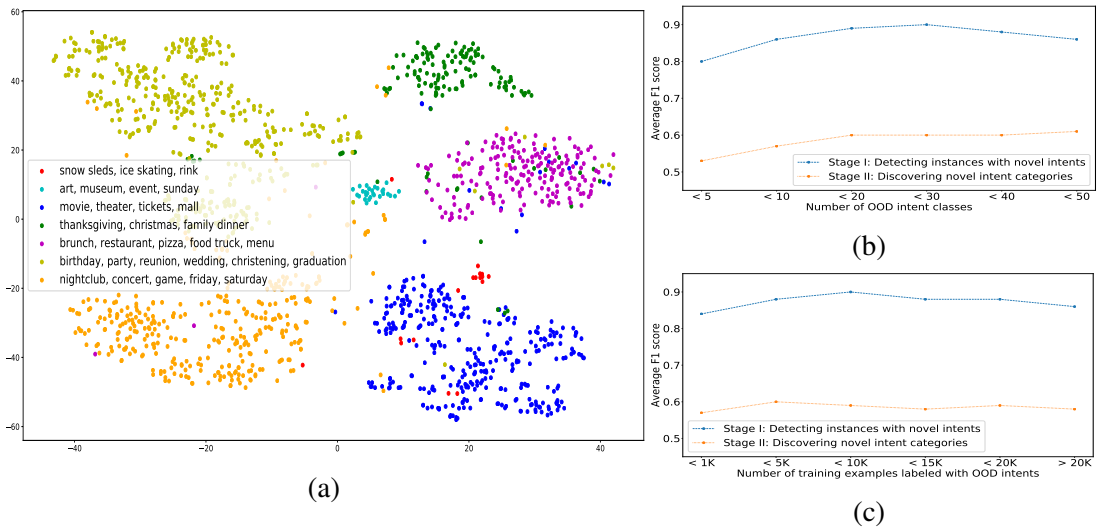
Ground truth intent and domain labels are available for the various data Sets we used for evaluation (see Table 6.1). We thus evaluate the novel intents and domains discovered via standard clustering metrics: (i) Comparing the number of discovered intents and domains to the ground truth number; (ii) Normalized Mutual Information (NMI): a normalization of the mutual information by a generalized mean of the entropy of the ground truth labels and the entropy of the predicted cluster labels; (iii) Purity: the extent to which a cluster contains utterances having the same intent or domain label; (iv) F1 score.

6.4.3 Results

Baseline Comparison: Table 6.2 shows the F1-score of various approaches, on classifying an intent as novel or not on different datasets with different dataset configurations. We observe that our proposed approach in the last row, using an $(S+m)$-class classifier and the DOC heuristic, significantly outperforms all baselines on all datasets by at least 6% F1 score points. ADVIN also outperforms the zero shot IntentCaps model (that uses relevant information available beforehand about the new test intents to be discovered), *without* using any prior knowledge about the novel intents to be discovered.

The last rows of Tables 6.3 and 6.4 show that Stage II of ADVIN as proposed significantly outperforms all baselines. The learned intent-clusters have a purity >75% and F1-score >60% for the SNIPS, NLUED and Internal datasets. Purity decreases to 61% for FTOP, primarily because semantically diverse utterances have the same ground truth intent label of *'unsupported'* or *'unsupported-event'*. The 'NA' value for "ADVIN *(ProdLDA)*" in the column '# int.' denotes that the *ProdLDA* algorithm does not output the number of new intents discovered. It takes this as input in the form of the number of topics. We find similar empirical trends for ATIS Sets 1 and 2, NLUED Set 2 and Internal Data Sets 2, 3 and 4. Due to lack of space, we do not present these results. In Table 6.5, we find that our proposed choice of hierarchical clustering in Stage II of ADVIN (last row), significantly outperforms a host of other clustering methods: X-means [241], hierarchical density based clustering (HDBSCAN) [210] and robust continuous clustering (RCC) [283].

ADVIN discovers 1.5-4.5 times more novel intents than present in the ground truth, as seen from the '# int.' columns in Tables 6.3 and 6.4. To investigate this further, we use t-SNE [200] to visualize the embeddings learned from Stage II of ADVIN, for the utterances belonging to the *'unsupported'* intent category of the FTOP dataset. In Figure 6.3a, each color represents a different novel intent category discovered. We also show the most frequent utterance words present in each novel intent. We observe that the *'unsupported'* category has been split up by ADVIN into 7 finer-grained, semantically sensible novel intents. For instance, the utterances *"What city has the most traffic in the US"* and *"Family friendly bars near me"*, from the *'unsupported'* intent category, have been separated by ADVIN into different intent categories. Figure 6.3a also explains the lower metric values in Table 6.3 for the FTOP dataset. We thus find that ADVIN largely learns semantically appropriate,



(a) (b) (c)

Figure 6.3: (a) Visualizing embeddings learned by Stage II of ADVIN for novel intents discovered within the *'unsupported'* FTOP category. We also show the varying F1 score averaged across (b) number of OOD intent classes and (c) number of labeled OOD instances, used to train Stage I, for the Internal data.

discriminative representations for the novel intents. This is also echoed by our user study in Table 6.6.

Effect of OOD training data: We analyze the effect of the labeled OOD instances given as training input to Stage I of ADVIN, i.e. detecting presence of novel intents in utterances. Figures 6.3b and 6.3c show that the performances of both Stages I (Section 6.3.1) and II (Section 6.3.2) of ADVIN improve with increase in the number of (i) OOD intent classes, and (ii) OOD training data examples; up to a particular threshold '*m*' for the Internal Data Set I. Increasing '*m*' beyond this value leads to a performance drop. We also observe that the presence of utterances with fine-grained intent labels (e.g. ATIS data) in the OOD data has a higher performance impact, than utterances with coarser-grained intent labels (e.g. FTOP data). We find similar trends for the other datasets also, but do not present them due to space constraints.

Table 6.6: Given an input pair of utterances, we evaluate ADVIN at predicting if both utterances contain the same novel intent or not. We show the F1-score computed via two sources of ground truth (GT) intent labels, separated by a '/': (i) a user study (US) and (ii) original dataset-provided intent annotations.

Approach	**FTOP: F1 US-GT / F1 Dataset-GT**	**Internal: F1 US-GT / F1 Dataset-GT**
ADVIN *(clf+hier)*	0.66 / 0.6	0.6 / 0.68
ADVIN*(triplet+hier)*	0.6 / 0.57	0.55 / 0.61
ADVIN *(ProdLDA)*	0.65 / 0.56	0.54 / 0.61
ADVIN*(proposed)*	0.7 / 0.61	0.61 / 0.71

Linking Intents to form Domains: There is no domain information available for the SNIPS, ATIS and FTOP data. Hence, we only evaluate the performance of linking intents into domains on the NLUED and Internal datasets. The column of '# dom.' in Table 6.4 for NLUED Set 1 and Internal Data Set 1 shows the number of newly discovered domains by ADVIN in Stage III (Section 6.3.3). As earlier, ADVIN obtains a finer-grained grouping of novel intents into novel domains. It finds 2-3 times more number of domains than the available dataset annotations. On further inspection, we found that the domains for both these datasets have been created by collating intents satisfying common overarching user needs, goals or motives, and are less geared towards semantics. Contrarily, ADVIN focuses more on the semantic meaning of the utterances, and performs a more fine-grained categorization of novel intents into domains. This leads to a slight discord between the domains uncovered by ADVIN, and the domain annotations in the dataset. Such an over estimation of the number of novel intents or domains can be often acceptable in a practical setting, since the granularity of domains and intents learned by ADVIN can be easily reduced or 'coarsened' by merging together certain novel intents or domains, based on downstream requirements. One way to do this is by soliciting human feedback, as we show in Table 6.8.

Table 6.7: Sample utterances present in the novel intents and novel domains discovered by ADVIN.

FTOP dataset (novel intents FI1, FI2, FI3)	**NLUED dataset (novel intents NI1, NI2, NI3)**
FI1: *Any parks near Fillmore that offer sledding; Do they have snow sleds at Ober Gatlinburg; Ice skate rink hours for Dec 9th* **FI2**: *Closest nightclub that has dancing; Find the nearest dance club that has a live band; Directions to the Die Antword concert* **FI3**: *Are there any drive-in movie theaters left in Ohio that are open in the fall; Movie theaters near me*	**NI1**: *Tell me about Mary S. in my contacts; how many numbers are saved for Ale; what is John Doe's address* **NI2**: *i need an email chain to my mother i'm planning a trip to see her ask her how the weather will be so i know* *how to pack; what is the subject of the email that just arrived; check for this mail in my contact if not then add it*
Novel domain FD1: novel intents FI2 and FI3	**Novel domain ND1**: novel intents NI1 and NI2

User Study: To compare the performance of ADVIN with respect to the human perception of novel intents and domains in greater detail, we conduct a *user study* on the FTOP and Internal NLU datasets. We recruit crowd workers on Amazon Mechanical Turk for the FTOP data and employees familiar with the Internal Dataset for this purpose. We provide a set of random utterance pairs predicted by ADVIN as having novel intents to the annotators, and ask them to indicate whether the pair is likely to belong to the same intent category or not. For both datasets we compute the F1-score in Table 6.6, by comparing the output of ADVIN with that of the human annotators, for 2500 FTOP utterance pairs (inter-annotator agreement Cohen's $\kappa = 0.78$) and 1100 Internal data utterance pairs (Cohen's $\kappa = 0.9$). We observe that ADVIN as proposed, significantly outperforms baselines by at least 5% with respect to human evaluation.

Case Study: We show in Table 6.7 some sample test set utterances for the FTOP and NLUED datasets (from Table 6.1), that have been discovered by ADVIN as containing novel intents, leading to the formation of novel domains. We observe that utterances talking about

Table 6.8: Performance of Stage II of ADVIN, with limited supervision input to hierarchical clustering in the form of pairwise constraints. Parentheses show original results of Stage II without using supervision.

Dataset	**# of novel intents discovered**	**NMI**	**Purity**	**F1 score**
SNIPS Set 1	2 (3)	0.82 (0.8)	0.94 (0.92)	0.85 (0.78)
ATIS Set 1	12 (13)	0.74 (0.72)	0.77 (0.73)	0.75 (0.72)
FTOP Set 1	17 (19)	0.5 (0.46)	0.64 (0.61)	0.54 (0.51)
NLUED Set 1	3 (5)	0.68 (0.63)	0.85 (0.8)	0.67 (0.64)
Internal NLU Data Set 1	45 (56)	0.65 (0.61)	0.75 (0.71)	0.65 (0.62)

similar or related topics (e.g. *sledding* and *ice skating* in the first row) are grouped into a single intent category. Utterances belonging to intents regarding 'music' (*dance clubs, concerts* in the second row) and 'movie' (third row) events are grouped together into a novel 'entertainment' domain. In the second column, utterances talking broadly about various *email* aspects are collated into a novel 'email' based domain.

Introducing Limited Supervision: There is often some partial supervision or background knowledge known to humans or provided by experts, regarding the unlabeled utterances. Utilizing this can enhance the quality of the novel intents and domains discovered by ADVIN, based on downstream requirements. Therefore, we test ADVIN in a semi-supervised setting, where we provide some prior knowledge to ADVIN in the form of pairwise constraints: (i) *must-link*, for utterances that must contain the same novel intent, and (ii) *cannot-link*, for utterances that cannot contain the same novel intent. We incorporate the constraints during hierarchical clustering, by modifying the learned distance values between the utterance pairs. In Table 6.8, we randomly select 3 groups of 4 utterances each (a total of < 25 utterances) for the *must-link* and *cannot-link* constraints respectively. This gives a 2-8% performance

gain across various metrics and datasets. This experiment shows that providing minimal supervision to ADVIN can significantly improve the quality of the discovered intents and domains over an unsupervised setting.

6.5 Conclusion

We propose a novel and generalizable framework, ADVIN, to discover novel intents and novel domains in large volumes of unlabeled text data, utilizing an existing labeled corpus of non-overlapping intents, a pre-trained language model and knowledge transfer. ADVIN significantly outperforms baselines on real datasets. In future, we plan to enable ADVIN to (i) detect multiple intents and/or domains per utterance; and (ii) use additional knowledge to better model human-perceived latent intents and domains.

Chapter 7: Multimodal Analysis of Digital Media Content in the Advertising Domain

We examined how to create structured sources of context from massive, noisy online information content to aid the study of latent pragmatic analysis in Chapters 2 and 3. In this Chapter, we now investigate the functional intent expressed in online media content and its connections with user reactions and behavior, by focusing on the advertising domain. In particular, we implemented a computational framework for the predictive analysis of the content-based features extracted from advertisement video files, to aid the design and production processes of commercial advertisements. Our proposed framework extracts multi-dimensional temporal patterns from the content of advertisement videos using multimedia signal processing and natural language processing tools. We employ a cross modality feature learning architecture where data streams from different feature dimensions are employed to train separate neural network models and then these models are fused together to learn a shared representation. Subsequently, a neural network model trained on this joint representation is utilized as a classifier for predicting advertisement effectiveness. Based on the predictive patterns identified between the content features and the effectiveness metrics of advertisements, we have elicited a useful set of auditory, visual and textual patterns that is strongly correlated with the proposed effectiveness metrics while can be readily implemented in the design and production processes of commercial advertisements.

7.1 Introduction

The recent integration of e-commerce infrastructures and web-scale multimedia distribution platforms has greatly increased the online presence of commercial advertisements and their impact on our society, while stimulating the development and deployment of innovative multimedia processing tools, content distribution schemes, and marketing behavioral models for digitally creating and disseminating persuasive advertisements with enhanced audience acceptance. Advertising along with product development, pricing and distribution forms the mix of marketing actions that managers take to sell products and services. Video advertisements airing on television and social media are a crucial link of attracting customers towards a product. However, the factors contributing to advertisement effectiveness are rather complex and are a focus of study in marketing science and consumer psychology.

In a landmark study, Lodish et al. [191] examined the sales effects of 389 commercials and found that in a number of cases advertising had no significant impact on sales. Many ideas of how to create effective advertisements come from the psychology literature [35, 67]. Positive or negative framing of an advertisement, the mix of reason and emotion, the synergistic interactions between music and narrative speech, the time arrangement of video shoots and the spatial organization, the type of message being delivered, the frequency of brand mentions, and the popularity of the endorser seen in the advertisement, all go into making an effective advertisement. Another area from which advertisers draw is drama. The use of dramatic elements such as narrative structure, the cultural fit between advertisement content and the audience are also important. But how these factors are combined to develop effective advertisements still remains a heuristic process.

The availability of large repositories of digital commercials, the advances made in content-based multimedia information retrieval, and the proliferation of user feedback/interaction mechanisms in social media, such as comments and like/dislike ratings provide us a new computational avenue for investigating the “success formula" of effective advertising.

In this work, each advertisement clip is first divided into a sequence of short segments, from which we extract multimedia visual and auditory features to model the temporal timeline of the content. We employ word-vector embeddings based on the text transcriptions of the online advertisements as the “semantic" textural features. These individual feature dimensions are utilized to train separate neural networks to produce high level embeddings in their respective feature spaces, followed by a model fusing stage that learns a multimodal joint embedding for each advertisement. We also analyzed the predictive patterns of advertisement effectiveness and have elicited a representative set of dependency patterns as a preliminary “effectiveness grammar". The novel methodological contributions of this work lie in the feature engineering and neural network architectural design. The primary, applied contributions of this work shed light on key questions of advertisement effectiveness from the related fields of multimedia, psychology, marketing, and television/film production.

7.2 Related Work

Previous work in targeted advertisement recommendation [356, 377] have utilized the visual and textual features of an advertisement along with user profile and click-through behavior for robust estimations of advertisement similarities and advertisement-viewer matching patterns. Content-based multimedia feature analysis is also crucial in the design and production of video commercials [74]. Multimedia features and their temporal patterns are known to show high-level patterns that mimic human media cognition and are thus

useful for applications that require in-depth media understanding such as computer-aided content creation [318] and multimedia information retrieval [178]. Besides signal processing and web content distribution applications, the use of temporal features is effective in media studies [38], movies [23] and music [104], because temporal shapes are easier to recognize and memorize for manual studies and are intuitive for computer aided explorations.

Our proposed framwork employs Long Short-Term Memory (LSTM) [123] and Deep Boltzmann Machine (DBM) [273] architectures to model the temporal structures in the content feature sequences, utilizing the flexible modeling capabilities of these neural network architectures for identifying patterns from hierarchical temporal resolutions and for modeling cross-dimensional dependencies. Related application of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been proposed in [383] to generate a vector representation for videos and "decode" it using an LSTM model. Sutskever et al. applied a similar approach in the task of machine translation [312]. Venugopalan et al. [350] used an LSTM model to fuse video and text data from a natural language corpus to generate text descriptions for videos. Deep Boltzmann Machines have also been used to model multimodal data in various fields such as speech and language processing, image processing and medical research [273, 299].

7.3 Methodology

The multimedia temporal features employed in our proposed framework are extracted from the video, audio, and text transcriptions of commercial advertisements. These content dimensions are selected due to their easy integrability into existing workflows of video post-production and script creation. The feature extraction stage is followed by learning a multimodal embedding for predictive modeling.

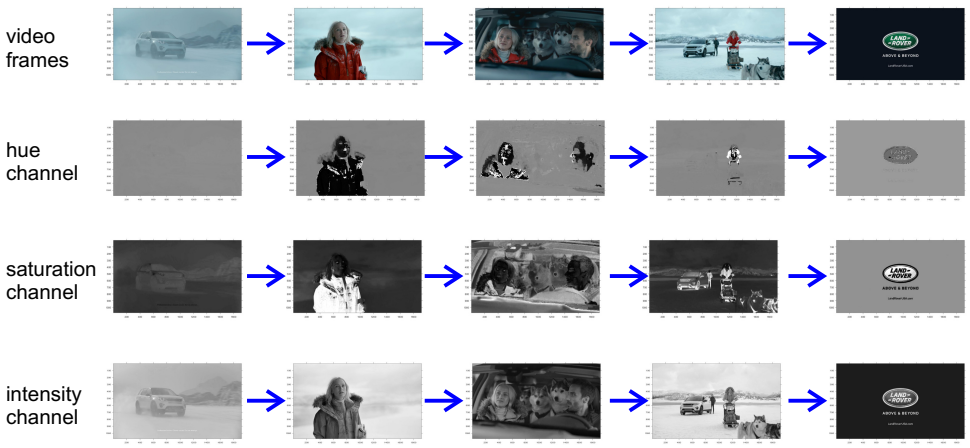


Figure 7.1: Multimedia timeline analysis of three video signal dimensions.

7.3.1 Feature Extraction

Visual/video Features: Video features of content timelines are extracted from the image features from sampled video frames as illustrated in Fig. 7.1. To speed up the signal processing algorithms, we sample one in ten video frames. We measure the hue, saturation and brightness values of each pixel in the sampled frames. The feature descriptors for each frame include the mean value and spatial distribution descriptors of the hue-saturation-brightness values of the constituent pixels. For measuring the deviations of these feature variables at different segments of the screen, the mean values of the screen's sub-segments and the differences between adjacent screen segments are calculated. We also segment the entire time duration of each video into 50/20/5 time segments as a hierarchical signal feature integration process and calculate the temporal statistics inside each segment including temporal mean and standard deviation, as well as the aggregated differences between adjacent frames.

Auditory/audio Features: Audio signal features include auditory loudness, onset density, and timbre centroid. Loudness is based on a computational auditory model applied on the frequency-domain energy distribution of short audio segments [221]. We first segment the audio signal into 100 ms short segments and calculate the fast Fourier transform for each segment for analyzing its time-frequency energy distribution. This short segment length setting ensures appropriate analytical resolution in both the time domain and the frequency domain. Because the human auditory sensitivity varies with frequency, a computational auditory model [218] is employed to weight the response level to the energy distribution of audio segments. The loudness L_a is calculated as:

$$L_a = log_{10} \sum_{k=1}^{K} S(k)\eta(k)$$

where $S(k)$ and $\eta(k)$ denote the spectral magnitude and the frequency response strength respectively at frequency index k. K is the range of the frequency component. The loudness feature sequence is then segmented and temporal characteristics like the mean and standard deviation in each segment are used as feature variables.

The audio onset density measures the time density of sonic events in each segment of $1/50^{th}$ of the entire video duration (typical segment length around 2 seconds). The onset detection algorithm [221] records onsets as time locations of large spectral content changes, and the amount of change as the respective onset significance. For each segment, we count onsets with significance value higher than a preset threshold and normalize the count by the segment length as the onset density. The timbre dimensions are measured from short time segments, similar to the loudness measurement above. The timbre centroid T_c for a short segment is calculated as:

$$T_c = \frac{\sum_{k=1}^{K} kS(k)}{\sum_{k=1}^{K} S(k)}$$

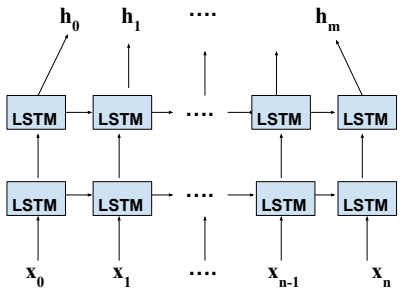


Figure 7.2: LSTM model with two hidden layers, each layer having 100 hidden units each, used for training individual input modalities.

Textual Features: Our proposed framework employs word2vec [213] for semantic textual analysis. Word2vec provides a robust approach to word vector embedding, using a two-layer neural network with raw text as an input, to generate a vector embedding for each word in its vocabulary. The textual feature analysis module in our implementation first extracts and preprocesses the text transcription of each advertisement to obtain word tokens. We then use the 300-dimensional vectors pre-trained on the Google News Corpus[14] to obtain word2vec token embeddings.

7.3.2 Learning Multimodal Feature Representations

LSTMs for Sequential Feature Modeling: We use an LSTM model with two layers to encode sequential multimedia features from Section 7.3.1, employing a model of similar architecture for all the three input modalities. We generate a visual feature vector for temporal video frames of each advertisement, which forms the input to the first LSTM layer of the video model. We stack another LSTM hidden layer on top of this, as shown in Fig. 7.2, which takes as input the hidden state encoding output from the first LSTM layer. Thus, the

[14] `https://code.google.com/archive/p/word2vec/`

first hidden layer would create an aggregated encoding of the sequence of frames for each video, and the second hidden layer encodes the frame information to generate an aggregated embedding of the entire video.

Next, we similarly encode the temporal audio feature sequence as a two hidden layer LSTM model. For the textual features, we first encode the 300-dimensional vector embedding of each word in the advertisement text transcription through the first hidden layer of an LSTM model. A second hidden layer is applied to this encoding to generate a summarized textual embedding for each advertisement.

Multimodal Deep Boltzmann Machine (MDBM): We employ the Gaussian-Bernoulli variant of the classical Restricted Boltzmann Machine [293], vertically stacking the RBMs to form a DBM [273]. Our implementation uses three DBMs to individually model the video, audio and text features. Each DBM has one visible layer $\mathbf{v} \in R^n$ with n units, and two hidden layers $h_i \in \{0,1\}^m$ with m units for $i = 1,2$. A DBM is an energy based generative model where the energy of the joint state $\left\{\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}\right\}$ is defined as:

$$P(\mathbf{v};\theta) = \sum_{\mathbf{h^{(1)}},\mathbf{h^{(2)}}} P\left(\mathbf{v}, \mathbf{h^{(1)}}, \mathbf{h^{(2)}}; \theta\right) = \frac{1}{Z(\theta)} \sum_{\mathbf{h^{(1)}},\mathbf{h^{(2)}}} exp\left(-E\left(\mathbf{v}, \mathbf{h}; \theta\right)\right)$$

$$E(\mathbf{v}, \mathbf{h}; \theta) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{ij} \frac{v_i}{\sigma_i} W_{ij}^{(1)} h_j^{(1)} - \sum_{jk} W_{jk}^{(2)} h_j^{(1)} h_k^{(2)} - \sum_j b_j^{(1)} h_j^{(1)} - \sum_k b_k^{(2)} h_k^{(2)}$$

where $\mathbf{h} = \left\{\mathbf{h}^{(1)}, \mathbf{h}^{(2)}\right\}$ denotes the units of two hidden layers and $\theta = \left\{\mathbf{W^{(1)}}, \mathbf{W^{(2)}}, \mathbf{b^{(1)}}, \mathbf{b^{(2)}}\right\}$ denotes the weights and bias parameters of the DBM model. $Z(\theta) = \int_{\mathbf{v}} \sum_{\mathbf{h}} exp\left(-E\left(\mathbf{v}, \mathbf{h}; \theta\right)\right) dv$ denotes the partition function.

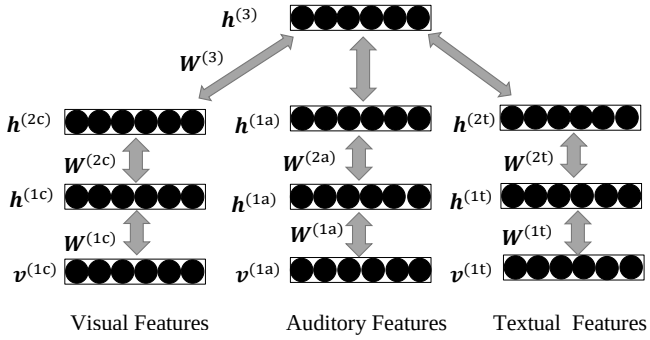


Figure 7.3: MDBM that models the joint distribution over the visual features, auditory features and textual features. All layers in this model are binary layers except for the bottom real valued layer.

Our architecture forms an MDBM configuration as in Figure 7.3 by combining the three DBMs and adding an additional layer on top. The joint distribution over the three input modalities is defined as:

$$P(\mathbf{v}^c, \mathbf{v}^a, \mathbf{v}^t; \theta) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} exp(-V - A - T + J)$$

where $\mathbf{v^c}$, $\mathbf{v^a}$ and $\mathbf{v^t}$ denote the visual, auditory and textual feature inputs over their respective pathways of V, A and T, J represents the joint layer at the top, and $Z(\theta)$ denotes the partition function. Here,

$$V = \sum_i \frac{(v_i^c - b_i^c)^2}{2\sigma_i^2} - \sum_{ij} \frac{v_i^c}{\sigma_i} W_{ij}^{(1c)} h_j^{(1c)} - \sum_{jl} W_{jl}^{(2c)} h_j^{(1c)} h_l^{(2c)} - \sum_j b_j^{(1c)} h_j^{(1c)} - \sum_l b_l^{(2c)} h_l^{(2c)};$$

$$J = \sum_{lp} W^{(3c)} h_l^{(2c)} h_p^{(3)} + \sum_{lp} W^{(3a)} h_l^{(2a)} h_p^{(3)} + \sum_{lp} W^{(3t)} h_l^{(2t)} h_p^{(3)}$$

A and T have similar expressions as V. $\mathbf{h} = \left\{ h^{(1c)}, h^{(2c)}, h^{(1a)}, h^{(2a)}, h^{(1t)}, h^{(2t)}, h^{(3)} \right\}$ denotes the hidden variables, $\mathbf{W}$ denotes the weight parameters, and $\mathbf{b}$ denotes the biases.

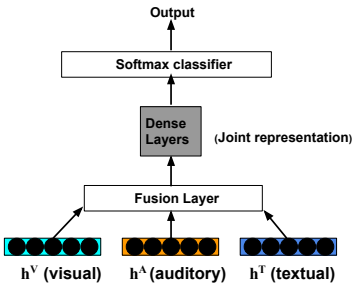


Figure 7.4: Fusing representations from the three individual modalities to infer a joint representation for an advertisment.

Inferring a Joint Representation: Once we obtain high-level feature embeddings ($\mathbf{h^V}, \mathbf{h^A}, \mathbf{h^T}$) from the final hidden layer of the three respective models of audio, video and text, we concatenate the three hidden layer embeddings in a layer called the *fusion layer*, which enables us to explore the correlation between the three kinds of features (see Figure 7.4). In order to minimize the impact of overfitting, we perform dropout regularization [298] on the fusion layer with a dropout probability of 0.5. The combined latent vector is passed through multiple dense layers with non-linear activation functions (rectified linear unit in our configuration), before being passed through a final softmax layer to predict the output class of the advertisement. We assume a binary classifier for the advertisements with two classes: effective or successful, and ineffective or unsuccessful. Thus, the probability of predicting a class label y is:

$$p(y|\mathbf{x_V}, \mathbf{x_A}, \mathbf{x_T}) \propto exp(W[\mathbf{h^V}; \mathbf{h^A}; \mathbf{h^T}] + \mathbf{b})$$

where y denotes the class, $\mathbf{x_V}, \mathbf{x_A}, \mathbf{x_T}$ are the video, audio and text features of advertisement x, W is the weight matrix, [;] denotes the concatenation operation and $\mathbf{b}$ the biases.

7.4 Evaluation

7.4.1 Dataset and Evaluation Metrics

We evaluated our proposed methodology on a dataset of 200 advertisements crawled from the *YouTube* website, spanning representative product categories such as food and beverages, clothing, consumer electronics, health and medicine, movie trailers, and service. The ground truth for whether an advertisement is successful/effective or not was based on three independent metrics. First, a user study was conducted on all advertisements comprising 900 responses to 96 questions on categories such as brand presence and reliability, emotion expressed in the advertisement, emotion induced in the user, attention paid while watching, its influence on the user etc. Most questions solicited ratings ranging from 'Strongly Disagree' (rated 1) to 'Strongly Agree' (rated 7). We considered advertisements with a mean rating less or equal to 3 (averaged over all questions) as ineffective, while the rest as effective. The rating results were anonymized, and the experiment content and procedures were approved by the Internal Review Board of the respective organizations involved (IRB number 2015B0249), and meet with standard Nielson Norman Group guidelines. Second, we scraped the users' *YouTube* comments on each advertisement, and calculated the strength of the sentiment expressed in them using a tool called SentiStrength [323]. The sentiment strength scores ranged from -5 to 5, and all advertisements having a mean score above a threshold of 2.5 were considered effective, and the rest as ineffective. Third, the number of 'likes' i.e. explicit positive feedback received by an advertisement video on *YouTube* is a clear indicator of its popularity. We calculate the measure of the likes an advertisement receives over its total number of views as a measure of its effectiveness. All advertisements

with a likes to views ratio above the mean of the ratio values received by all advertisements were categorized as effective.

7.4.2 Results

We compare our method against the baseline classifiers of Linear SVM and Logistic Regression, which take as input a concatenation of the visual, auditory, and textual features (Table 7.1). We trained our neural network models over 15 epochs, minimizing the binary cross entropy loss using the Adam [160] optimizer, with a learning rate of 0.001. We randomly selected 150 advertisements for training and 50 for testing our method, and averaged our results over 50 runs. Compared to our other models, *the multimodal LSTM model achieved the best accuracy and an F1-score greater than* 0.8*, and the difference in accuracy is significant*. It also has a false positive rate of 0. Using a multimodal joint feature representation gives a huge advantage over any of the individual models. The text-only LSTM model that classifies advertisements based only on textual features appears to perform better than the video-only and audio-only models, whose accuracies are below 50%.

In addition, we removed all occurrences of brand name from the advertisement text and found the accuracy of the Text-only model to reduce to nearly 46%, while the accuracy of the multimodal LSTM dropped to about 73%. This confirms that the presence of brand name is crucial in determining advertisement success. We also inspected the impact of the position of the brand name in the advertisement text i.e. its occurrence at the start, middle or end of the advertisement, but did not find any significant difference in performance.

Regardless of the evaluation measure adopted, the multimodal LSTM model significantly outperforms the other models, followed by the multimodal DBM model. Hence, for the purpose of evaluation and model selection we hypothesize that one can employ metrics

Table 7.1: Classification results using various classifiers and ground truth metrics (best performance in bold)

Model	**Ground truth**	**Accuracy**	**F1**
Linear SVM	Comment sentiment	0.58	0.565
Linear SVM	Likes/visits	0.586	0.568
Linear SVM	User study	0.565	0.541
Logistic Regression	Comment sentiment	0.468	0.44
Logistic Regression	Likes/visits	0.55	0.529
Logistic Regression	User study	0.542	0.52
Multimodal DBM	Comment sentiment	0.60	0.66
Multimodal DBM	Likes/visits	0.61	0.71
Multimodal DBM	User study	0.66	0.64
Multimodal LSTM	Comment Sentiment	0.786	0.765
Multimodal LSTM	Likes/visits	0.8	0.769
Multimodal LSTM	User study	**0.83**	**0.81**
Video-only LSTM	Comment Sentiment	0.34	0.334
Video-only LSTM	Likes/visits	0.39	0.378
Video-only LSTM	User study	0.44	0.408
Audio-only LSTM	Comment sentiment	0.365	0.341
Audio-only LSTM	Likes/visits	0.401	0.4
Audio-only LSTM	User study	0.416	0.37
Text-only LSTM	Comment sentiment	0.478	0.445
Text-only LSTM	Likes/visits	0.49	0.468
Text-only LSTM	User study	0.52	0.52

derived from easily available online information such as likes, views and comments, rather than opting for the more expensive method of performing a user study. Having said this, we note that using the user study as ground truth shows a statistically significant performance improvement. For instance, the difference in accuracy between the multimodal LSTM models evaluated on the user study and on the ratio of likes per visits has a p-value of 0.04123 whereas the difference in F1-score between the two has a p-value of 0.0133, using a McNemar's paired significance test.

In addition, we seek to study the mapping relationships between multimedia attributes and advertisement success. For this purpose we choose a random forest classifier (yielding a

classification accuracy of 0.55). In case of the visual attributes, the average intensity and average chroma for the first and second video segments are found to be important. The average saturation span and average chroma span for the fifth spatial zone i.e. the central zone of the screen have also been recognized as essential. In case of audio features, we obtain as important the onset spectrum strength, onset spectrum variation, and onset density dynamic range for the second and penultimate audio partitions.

In order to validate the importance of the above features, we performed experiments using our proposed model after excluding these particular audio-visual features from the input and using the user study as a ground truth metric. The textual input remained the same as earlier. We found a significant reduction in classification accuracy for the LSTM model, down to about 67%, while the accuracy of the DBM model went down to about 61%. We then utilized just the top important audio-visual features and the entire textual feature set as the input data. The accuracy of both models was found to reduce to 70% and 63% respectively, with the reduction also possibly due to loss of information via feature elimination. However, using only the important features still yields a reasonable classification accuracy. Thus, the above identified video and audio features are indeed essential in identifying and characterizing effective advertisements.

7.5 Discussion

Our findings show that the video segments in the first few seconds of an advertisement significantly mark out effective advertisements from ineffective ones. This concurs with the classical interpretations from marketing literature [191], which argue that viewers pay very little attention to details while watching advertisements, and thus effective advertisements must first attract the target viewer's attention before delivering its main message. Our

results also show that audio features of the second audio partition predicts advertisement effectiveness; agreeing with marketing literature that stating relevant music can grab viewers' attention [30]. However what is new in our results is that the order in which the visual and auditory elements occur in an advertisement attracts and holds viewers' attention.

Our finding that the video features of the central part of an advertisement is important for its effectiveness suggests that this is where the core message of the advertisement is embedded. We also find that brand mentions in an advertisement makes it more effective. This result finds support in the branding literature [39]. However, our finding that the temporal location of brand-mention is irrelevant is not supported by marketing literature. Scholars report that it is better to convey brand names and logos after the introductory attention-grabbing phase. We offer two reasons for our results. First, we do not control for the product category effect. For example, the temporal location of brand mention may vary based on if the advertisement is about a drug or a vacation. Second, we do not control for the brand's stage in its product life cycle. That is, where the brand name and logo appears may matter differently if we are dealing with a new brand or a mature brand. Our small sample size did not permit this analysis.

One unexpected finding from our predictive analysis is that text features explain advertisement effectiveness more than either video or audio features on average (see Table 7.1). Recall that our text features come from the transcription of the voice-over in the advertisement. This means that viewers prefer advertisements in which brand benefits are conveyed verbally and with which they can connect emotionally. This finding concurs with the argument that though visual and auditory information grab our attention quickly, verbal information is cognitively demanding, forming strong associations in our brain because when we hear words we spontaneously extract meaning from them [151].

In summary, our findings show that effective advertisements in our dataset exhibit strong creative design and production cues in visual, auditory and linguistic dimensions. Yet, they can be computationally analyzed and replicated in a rational pattern analysis process. Targeting audience acceptance and persuasive effectiveness, and creating effective advertisements demands strategic allocations of feature patterns from video, audio and textual script dimensions to form persuasive narrative patterns, e.g., first draw the viewer's attention in one content dimension and then deliver the brand message through another content dimension.

7.6 Conclusion

We implemented a computational framework for modeling the temporal patterns of visual, auditory, and textual features extracted from multimedia content of online advertisement videos, and explored the correlations between these content features and various subjective metrics of advertisement effectiveness. The temporal patterns in multiple feature dimensions are modeled by three individual neural network models, which are fused together to yield a joint embedding for representing the pattern dependencies between different feature dimensions for predictive modeling and analysis. A binary softmax classifier is then employed to predict advertisement effectiveness. The predictive performance of our approach was validated using subjective ratings from a dedicated user study, the text sentiment strength of online comments and the likes/views ratio from *YouTube* web platform. The strong predictive performance obtained from the LSTM variants we proposed seem in line with its recent success in multimedia information retrieval and natural language processing [312, 350].

Chapter 8: Predicting Trust Relationships in Online Users

Our work described in the previous Chapters relies heavily on content generated by human users. Therefore, the trustworthiness or credibility of sources from which new information often arises is a key task to explore. In this Chapter, drawing on social and psychological theory, we detect pairwise and global trust relations between social media users in the context of emergent real-world crisis scenarios. In such situations and scale, seeking explicit pairwise trust assessments between online users is impractical. Instead, in an unsupervised manner, we integrate the implicit factors of social influence exerted by each user over the network, the underlying network structural topology and the affective valence expressed by the users in the textual content they communicate. A key finding is the importance of modeling influence and affective valence in such exchanges and their role in detecting stable trust relationships. We extensively evaluate these ideas and demonstrate significant gains over competitive baselines across multiple datasets drawn from both crisis and non-crisis scenarios, including those with normative ground truth.

8.1 Introduction

Trust is a vital social construct that has drawn attention from multiple areas of research including sociology, psychology, management, economics, political science and computer science. With the rise of Web 2.0 technologies, trust has emerged as a key concept in social

network and social media analysis, reflecting credibility and reliability for the multitude of online participants and data [82]. In particular, wireless technologies and social networking tools (e.g. Facebook) have created *citizen sensing*, appearing in CNN iReports and Twitter event commentary. Emergency response provides an ideal domain for examining citizen sensing [42]. Citizen response can have an enormous influence on the societal impact (cost, recovery) of a disaster. Victims and their neighbors share timely information (e.g. flood level, road blockages) and offer resources (e.g. vehicles, food and supplies) on social media, leading to the prioritization of relief effort ranging from critical infrastructure repair to saving lives in affected areas. Affordability, reach, proximity and timeliness make social media such as Twitter an attractive resource for capturing public observations and activity during emergencies [207]. However, message recipients such as response agencies must trust citizen sources to provide reliable information, and must rapidly vet and separate noise (inaccurate information from unreliable sources or ambiguities from reliable sources) from the informative signal.

Our primary focus here is on trust among users as opposed to trust in *facts*, e.g. road closures. We adopt the notion of interpersonal trust proposed by Kelton et al [152] who consider dyadic trust as a social tie between two individuals. We only consider information that is available or can be inferred from the social network and do not take into account any detail regarding background or previous history of entities. We develop an unsupervised algorithm to predict user trust relationships in social networks, using both structural properties of the network and information content posted by members of these networks.

A key contribution of our work is a robust method that takes into account indicators of influence as a proxy for how much one user trusts another: structural cohesion within a network, and valence-enhanced content (sentiment expressed related to the concept being

sensed). Taking valence into account provides a robust and stable model of trust in such an analysis. To the best of our knowledge, this is the first unsupervised effort taking into account social theories related to influence, passive cohesiveness (structural cohesion) and active cohesiveness (valence and content) and their role on inter-personal trust. We show that our method detects pairwise and global user trust relations on a range of real-world crisis and non-crisis datasets and outperforms extant state-of-the-art methods. We also find it effective in *dynamically* identifying trustworthy users in situations of crisis or emergency.

8.2 Related Work

Existing literature addresses the challenge of identifying or predicting the credible aspects of data and entities, under a social model of trust [321]. A simple strategy to encode a pre-defined notion of trust involves learning the encoding (via regression or classification) from a labeled set of specific interactions and their associated values obtained from a domain expert. Subsequently the learned model automatically categorizes the trust values associated with other relationships within the network. Both the biological and social network analysis literature rely on the local topology within the network to assess the confidence associated with a specific relationship [84]. However, topological signals alone may be insufficient. For example, Palen et al [351] imply that content and context, and metrics such as re-tweeting by others reflect a notion of trust between the user and her immediate followers. Some of these signals can be directly recovered from the data (e.g. location-specific tags from Twitter, volunteered GIS information etc.) but others must be inferred through suitable content analysis and other methods [224].

We have hypothesized that the idea of expertise or influence can serve as a prominent contributor to trust. Influence can often be estimated from the structure of the global

network, as in Twitter-Rank [363] and Trust-Rank [367], or the local network via viewpoint-based methods [9]. A large body of work has focused on studying diffusion and influence specifically on Twitter, using measures of influence such as number of followers, page-rank, number of retweets and number of mentions [43, 168, 363]. The influence-passivity algorithm by Romero et al [265] which we employ in this work also makes use of Twitter retweets and URLs, however it also simultaneously accounts for the passivity of network users as well, which can be important as explained in Section 8.4.

Social influence coupled with contextual information has been used in the literature in the problem of influence maximization [8, 16, 52, 153], where the objective is to identify seed users such that the number of users they influence in the network is maximized. We note that though influence plays a crucial role in our algorithm in quantifying trust, our work is distinct from these efforts in that we do not aim to study influence (and consequently, trust) propagation through the network or find the set of users with the maximum overall trust reach. We study both pairwise trust relations among users as well as identify users who are globally trusted. Having said that, we do compare our work against the topic-aware influence maximization algorithm by Aslay et al [8] and find that along with influence and context, shared content and valence information play a non-trivial role in discerning trust among users (Section 8.5).

One may not have enough data or domain knowledge to assess a meaningful trust value for all of the entities and relationships in the network. Hence, propagating trust values across entities using a set of rules [107] may be essential to obtain a completely specified trust network. Golbeck and Hendler [97] describe some of the challenges with trust propagation (including conflict resolution) based on the underlying model of trust and highlight the following properties regarding trust propagation: asymmetry (person A may

trust person B but not vice-versa), transitivity (if person A trusts person B and person B trusts person C, a trust relationship can be inferred between A and C) and composability (a user should aggregate trust values received from different paths). The TidalTrust algorithm [96] propagates trust ratings along a path between a source and a sink in Friend-Of-A-Friend based social networks using a Breadth-First search. EigenTrust [147] and MoleTrust [208] are peer-to-peer algorithms for determining peer trustworthiness.

A crucial dimension to the study of trust is the fact that trust is context-dependent; different users may be trusted differently on different contexts or topics. Tang et al [320] in their work on mTrust develop a topic-specific tensor representation in which a user probabilistically trusts another on a certain topic. In our work, we incorporate contextual information as a contributor to pairwise user trust scores but we do not currently compute topic-specific user trust values. We also utilize the idea that similarity in emotion expressed by users can be a crucial signal of the trust between them, which has also been studied by Beigi et al [18].

Finally, the notion of structural similarity between users has a role to play in shaping their trust relationships. Tajfel [314] defines the concept of social identity as “the individual’s knowledge that he belongs to certain social groups together with some emotional and value significance to him of this group membership". Such shared identity among members of a group can in turn lead to cohesiveness, uniformity and the motivation to sustain the reputation of their associated identity, which can consequently increase the feeling of mutual trust and affinity among the group members. In the literature, Golbeck [95] supports the idea of a positive correlation between user profile similarity and personalized trust values, while Yeung et al [11] find similarities in user trust networks on product review sites. Tang et al [319] model the effects of homophily in trust prediction with a low rank matrix

factorization model. We incorporate this concept into our trust calculation by increasing the pairwise trust score between users if they belong to a “group", i.e. if they share a common context.

8.3 Problem Formulation

Formally, we assume the social network of interest is represented as a bipartite graph $G = (U \cup T, E)$, where U and T are two disjoint sets and E is the set of edges or interactions between them. U represents the set of individuals or users in the network and T refers to the set of general topics expressed by the users in their textual content. Topics are extracted after aggregating the text posted by the users, by employing standard topic modeling algorithms based on Latent Dirichlet Allocation [22] or Non-Negative Matrix Factorization [373]. Our algorithm aims to predict the pairwise trust value between every pair of users, and generate an aggregated rank ordering of network users based on this trustworthiness score.

8.4 Methodology

We develop a model to infer trust among users within a social network, and identify highly trusted users or organizations in the network. We focus on the social network *Twitter*. Our solution desiderata include the goals of efficacy and efficiency: Efficacy in effectively identifying and ranking trustworthy users and efficiency in being able to compute these in real-time emergent situations (e.g. during or shortly after a disaster). We discuss below three elements informed by social and psychological theory, that we believe play a role in developing trust between two users in a social network. We measure these elements to determine a trust metric among users.

8.4.1 Influence

Theory: Leading sociologists note that trust is integral to social influence [76, 172, 313]. A messenger more easily influences or persuades a recipient to respond in a certain manner if that recipient trusts the original messenger. While trust itself is difficult to measure outside of specific experimental paradigms (for instance Berg's trust game [20], which addresses this problem from the perspective of behavioral economics), several researchers have tackled the problem of detecting *influential* users within a social network.

Here we exploit influence as a proxy for pairwise trust relationships. Theory suggests a strong correlation between the trust of a user x on a user y and the observed influence of user y on user x. Intuitively, users trust the users who influence them to re-post (or retweet) a particular post or tweet on a certain topic. The available measures of influence include the use of the structure of the network such as page-rank style [235], the dynamics of the network interaction [70], the frequency with which the users' posts are re-posted while accounting for user passivity and prior content history of the users' posts [265] or by taking into account the local neighborhood of the posting user via viewpoint analysis [9].

Implementation: Guided by the above theory and the available measures of social influence, we settled on the Influence-Passivity algorithm [265]. It is simple and also accounts for user passivity i.e. the likelihood of a user reacting to a messenger. Passivity is an essential constituent of the influence computation because intuitively, the retweeting or re-posting of a user's tweet should carry more weight from a typically passive user (who rarely retweets) than from a relatively active or frequent retweeter. The resulting HITS-style algorithm [161] calculates a global influence and passivity score for each user.

We first construct a weighted, directed, unipartite graph $H = (U, E_h, W_h)$ consisting of all the users in set U of our bipartite graph described above, joined by a set of edges E_h and

a set of edge weights W_h. Edge (i, j) exists between a user i and user j in the graph if user j re-posts or shares in some way (or retweets) a post (or tweet) posted by user i at least once. Weight w_{ij} on edge $e = (i, j)$ represents the ratio of influence that i exerts on j to the total influence i attempted to exert on j. It is expressed as $w_e = \frac{S_{ij}}{Q_i}$ where Q_i is the number of posts that user i creates and S_{ij} is the number of posts i created and j re-posted or retweeted.

The influence function $Infl_i : U \rightarrow [0, 1]$ that represents node i's influence on the network is calculated as:

$$Infl_i = \sum_{j:(i,j)\in E} u_{ij} Passiv_j$$

$$Passiv_i = \sum_{j:(j,i)\in E} v_{ji} Infl_j$$

$$u_{ij} = \frac{w_{ij}}{\sum_{(k,j)\in E} w_{kj}} \text{ and } v_{ji} = \frac{1-w_{ji}}{\sum_{(j,k)\in E} (1-w_{jk})}$$

Here u_{ij} represents the amount of influence user j accepted from user i normalized by the total influence j accepted from all users in the network. v_{ji} represents the influence that user i rejected from j normalized by the total influence rejected from j by all users in the network.

8.4.2 Social Cohesion

Theory: Social cohesion theory posits that a necessary and sufficient condition for individuals to work as a group is cohesive social relationships among individuals within the group. While social relationships exist for different reasons (e.g. kinship ties or similar social values), we focus on a group's structural cohesion, the collective result of those various social relationships. Sociologists and public policy experts believe that if common values, *trust* or a shared sense of place emerge, they will do so as a function of structural (cohesive) engagement [172]. Similar ideas are also posited by Golbeck and Ziegler et al [95, 402].

We adopt the definition by Lott and Lott [193] that interprets cohesiveness as a function of interpersonal attraction between individuals. In other words, cohesiveness relates to the members of a group who share emotional and behavioral characteristics with one another and the group as a whole [193, 255].

Implementation: We compute the Jaccard similarity metric [46, 93] between the neighborhoods of pairs of users in the bipartite network. This approximates the number of triads each pair belongs to, or the local triangle density. The notion of triads has been widely used to characterize community cohesiveness within a network [98, 105]. Such measures are also easy to approximate efficiently [277]. Based on the above theoretical insights, two users can be said to belong to a 'group' or possess some similar behavioral characteristics if they share common contextual interests. Specifically, we associate with each user x a *vector* V_x of topic identifiers to which x has a directed edge i.e. all topics on which a user shares text online. The Jaccard similarity between the users x and y is then:

$$Jacc(x,y) = \frac{|V_x \cap V_y|}{|V_x \cup V_y|}$$

In measuring shared topic interests among users we rely on what the social science literature refers to as passive cohesive interactions [172, 193]. We next examine the role of active relationships, which account for specific content, popularity of said content and the emotional attachment (sentiment/valence) associated with such content by actors within such networks.

8.4.3 Valence

Theory: We hypothesize that a critical dimension in the trust equation is the positive interactions, i.e. supportive exchanges among individuals and more broadly among communities, or "active social relationships". Such contacts and connections reflect trust as they offer people

and organizations mutual support, information and credit. This is particularly relevant during times of need; such as those that arise in crisis and disaster settings [172]. The Jaccard measure just described largely reflects similarity between users in the social network based on structural cohesion. It does not account for the popularity of certain topics that the users talk about, nor as suggested by Lott and Lott [193] does it account for emotional content (valence ranging from positive to negative) and interpersonal agreement among users. We note that to the best of our knowledge, while the corresponding social theory exists, none of these have been examined in the context of online social networks. To overcome the limitations of the Jaccard measure, we propose a more nuanced measure that accounts for the relative *popularity* of the topics that the tweets or text of the users primarily describe, the valence of a user with respect to a particular topic and the agreement among them.

Implementation: We introduce the notion of a *shared* topic (or context) between two users in U as any topic in T to which both users in graph G have a directed edge. Our implementation accounts for a number of confounding influences on apparent sharing. We distinguish between shared topics according to their relative popularity, i.e their in-degree in the bipartite graph. Intuitively, as the in-degree of a topic (the number of users talking about it) increases, the extent to which we can infer about the relative similarity of two users who post on that topic reduces. Two users who post on a rare topic are more likely to have an affinity or an interest towards that topic and consequently towards each other and therefore, a stronger trust relationship. Nevertheless, a pair of users posting on a popular topic may be doing so precisely because of its popularity or it being the current trend (see Figure 8.1(a)). User baseline activity level is also relevant. If two users post frequently on numerous or diverse topics, sharing a topic may not attest as much to their similarity (see Figure 8.1(b)) as for a pair of less active users sharing a common topic. Figure 8.1(c) thus

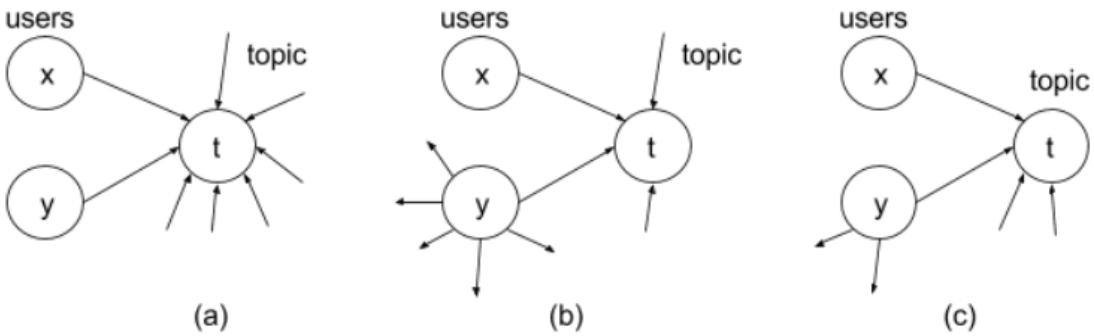


Figure 8.1: Content based user similarity, based on degree discounting

represents the case in which the pair of users x and y have the strongest likelihood of being similar to each other (and consequently trusting each other) based on their textual content. We now formalize the notion of *degree discounting* to capture the above ideas as follows:

1. When two users x and y share a topic t, the contribution of t to the similarity between x and y is inversely related to the in-degree of t.

2. The out-link similarity between two users x and y is inversely related to the out-degrees of x and y.

$$sim_d(x,y) = \frac{1}{\sqrt{D_o(x,x)}\sqrt{D_o(y,y)}} \sum_t \frac{A(x,t)A(y,t)}{\sqrt{D_i(t,t)}}$$

$sim_d(x,y)$ is the degree-discounted out-link similarity between users x and y. A is the adjacency matrix of the bipartite graph G. D_o is the diagonal matrix of users' out-degrees and D_i is the diagonal matrix of topics' in-degrees in G. t represents a shared topic between x and y, and can be defined in two ways: i) *without valence:* the pair of users merely post text or tweets on t, and ii) *with valence:* they not only post on t but also express the same dominant valence towards it (positive, negative or neutral) in their text.

The inclusion of valence expressed by the users on a particular topic lends more validity to pairwise user similarity and may reflect greater trust between a pair of users, reflecting the theoretical notion of interpersonal emotional agreement. A user x likely has greater trust in a user y who shares x's opinion on a particular event or topic. To quantify the overall opinion or valence within the textual content expressed by each user on each topic, we use a widely used tool called SentiStrength [323]. It builds a simple discriminative predictor based on psychological theories of emotive strength, exploiting de-facto grammar and spelling styles (e.g. informal spellings, social media specific linguistic terms, idioms and emoticons). We found this tool to be more effective than traditional ones such as LIWC [244] and ANEW [28] that are better suited to longer and more linguistically standard pieces of text.

8.4.4 Putting It All Together

To combine the three measures or quantities (influence, cohesion and valence), we first did a systematic study across a range of datasets described in Tables 8.1 and 8.2. We observed that each of these measures are well estimated by a Gaussian distribution suggesting that a natural way to standardize these quantities is to employ z-score normalization [165]. Once each measure is normalized we determine the regularization coefficients (via a parametric grid search as described in the next section) employed in the overall trust equation as follows:

$$Trust(x,y) = \alpha Infl(y,x) + \beta Jacc(x,y) + \gamma sim_d(x,y)$$

$Trust(x,y)$ represents the pairwise trust of user x on user y. α, β and γ are regularization parameters representing the factor weights and are tuned empirically such that $\alpha + \beta + \gamma = 1$.

After computing the value of pairwise trust for every pair of users, we assign a global trust score $GTrust(x)$ to each user x in the following way:

$$GTrust(x) = \frac{\sum_{y \in T_x} Trust(y,x)}{|T_x|}$$

Here, T_x denotes a set of users that have a non-zero pairwise trust score with user x, and $|T_x|$ denotes the cardinality of set T_x. Based on these global trust scores assigned to users, we generate a trust-based ranking of all the users in the dataset. The relative order of *GTrust* values provides a notion of ordinality regarding the overall trust a user enjoys within these network contexts. Depending on the values of α, β and γ employed to compute pairwise trust, we obtain multiple distinct trust-based user rankings.

Complexity Considerations: To compute a trust score between all pairs of users in a social network, an $O(n^2)$ complexity is unavoidable. Particularly for larger datasets with millions or billions of users, computing trust between every pair of users in the network will not scale, so pairwise trust might need to be only computed for a sample of users. For discovering the top trustworthy users one can potentially employ a smart sampling strategy where the trust for each user is accumulated over a sample of other users that interact with that user. Strategies often employed in similarity search problems [46, 93] can also be leveraged.

In our algorithm, the complexity of computing individual factors is as follows: Computing influence has the same complexity as the HITS algorithm [161] i.e. $O(E)$ per iteration; in our experiments we typically converge in less than 30 iterations. The complexity of determining the structural cohesion of two users as well as the valence step is bounded again by $O(E)$ (although in practice it is much less with our proposed optimizations). The complexity of parametric tuning (for each of the factors of influence, cohesion, and valence) can potentially be expensive. However, as discussed next, our results strongly suggest a robust weighting scheme highlighting the importance of influence, followed by content valence, followed by passive social cohesion.

8.5 Evaluation

8.5.1 Datasets and Ground Truth

The datasets we employ in this study are described in Tables 8.1, 8.2 and 8.3. All social media data was collected using Twitter's Streaming API. Because ground truth for each pairwise trust relationship from social media data is difficult to obtain without a dedicated social survey instrument, we also include results on two datasets with normative ground truth: i) a Film DVD review dataset collected by Guo et al [108] in December 2013 by crawling 17 categories of film DVDs from the `dvd.ciao.co.uk` website; and ii) a dataset made available by Massa et al from the Epinions product rating and review website [209]. Both these datasets provide authentic ground truth information as a list of pairs of users who trust each other (scaled values from 1 to 5). We construct a bipartite graph for these two datasets as described in Section 8.4. Because there is no direct measure of re-posting or retweeting associated with these two datasets, for the purpose of computing influence we assume that a user x functionally 'retweets' a user y if x rates $y's$ reviews more than three times. For contextual information, we used the provided genre of each film and the provided category of each product respectively, as topics. Additionally, we used user provided ratings to estimate the user valence towards a particular film or product. A user rating of 4 or more (out of 5) was taken as the expression of positive valence, 2 or less was taken as negative valence and a rating of 3 was assumed to be neutral.

8.5.2 Factor Analysis and Impact of Valence

Metrics: We begin by comparing our global trust-based user ranking (details in Section 8.4.4) at various parameter value settings against the ground truth user trust ranking for the CiaoDVD and Epinions datasets. This provides an independent assessment of our

Dataset	#Users	#Topics	#Nodes	#Edges	#Tweets
India Anti-Corruption	2104	15	2119	7180	100K
Mumbai Blast	581	10	591	932	10K
Phone and Tablet	9939	15	9954	16265	100K
Houston Floods	986	15	971	875	100K
Hurricane Irene	50561	20	50541	17593	200K
Nice Attack	253243	20	253223	166943	800K

Table 8.1: Bipartite graph statistics of social media datasets

Dataset	#Users	#Topics	#Nodes	#Edges	#Trust relations
CiaoDVD	7375	17	7392	111781	40133
Epinions	114467	27	114494	442787	717667

Table 8.2: Bipartite graph statistics of non-social media datasets with ground truth trust

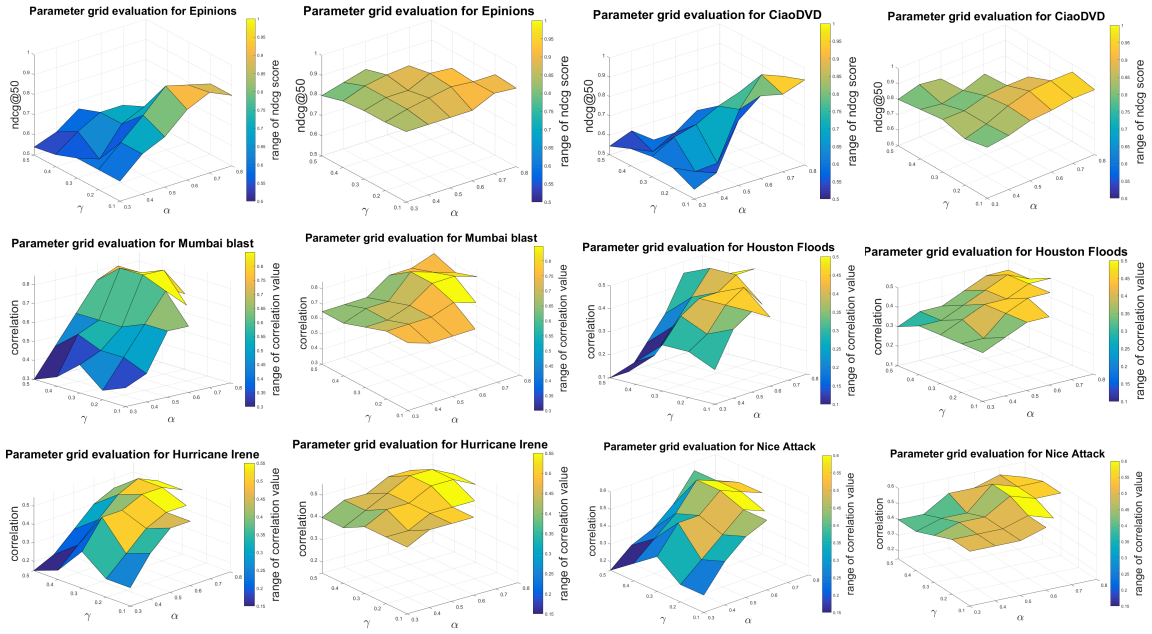
Dataset	Timeline
India Anti-Corruption	April 4, 2011 to December 28, 2011: Protests against political corruption in India (`https://en.wikipedia.org/wiki/2011_Indian_anti-corruption_movement`)
Mumbai Blast	July 13, 2011: A series of three coordinated bombings in Mumbai (`https://en.wikipedia.org/wiki/2011_Mumbai_bombings`)
Hurricane Irene	August 21, 2011 to August 30, 2011: A tropical cyclone on US East Coast that grew into a hurricane (`https://en.wikipedia.org/wiki/Hurricane_Irene`)
Phone and Tablet	All tweets related to phones and tablets, during the day of April 15, 2013
Houston Floods	April 15, 2016 to April 23, 2016: Snowstorm leading to severe floods (`https://en.wikipedia.org/wiki/2016_Texas_floods`)
Nice Attacks	July 14, 2016: A cargo truck driven into a crowd, followed by a gunfire which led to an emergency (`https://en.wikipedia.org/wiki/2016_Nice_attack`)

Table 8.3: Timeline and details of each Twitter dataset

method on datasets with factual ground truth. For this, we made use of the standard information retrieval metric of Normalized Discounted Cumulative Gain (NDCG) [135]. NDCG is often the measure of choice for comparing ranked lists since it emphasizes high fidelity with

Dataset	α	β	γ
India Anti-Corruption	0.6	0.05	0.35
Mumbai Blast	0.65	0.05	0.3
Phone and Tablet	0.7	0.2	0.1
Houston Floods	0.65	0.1	0.25
Hurricane Irene	0.65	0.05	0.3
Nice Attacks	0.7	0.05	0.25
CiaoDVD	0.7	0.1	0.2
Epinions	0.65	0.1	0.25

Table 8.4: Best parametric settings for different datasets



(a) without valence (b) with valence (c) without valence (d) with valence

Figure 8.2: The first row of sub-figures shows the NDCG scores of various rank orders for the parameter grid against ground truth for CiaoDVD and Epinions. The next 2 rows show correlation between the pairwise trust score and conversation length for the parametric grid for social media datasets; in both cases using content-based similarity **without and with valence**

ground truth ranking at the top of the list, which aligns with our goals for identifying top trusted users in emergent situations. The NDCG score at a rank position k (we used $k = 50$ in our experiments) is defined as:

$$DCG_k = \sum_{i=1}^{k} \frac{2^{imp_i}-1}{log_2(i+1)}$$

$$NDCG_k = \frac{DCG_k}{ideal_k}$$

where imp_i is an integer (between -2 and 2 based on the position of the user in the original rank list) representing the importance of user i, DCG_k is the discounted cumulative gain at rank k and $ideal_k$ is the ideal ground truth ranking.

For social media data, ground truth information on pairwise trust is difficult to obtain, which requires us to identify an approximation to ground truth. Adali et al [1,2], recommend the use of the length of conversation between pairs of users as an approximate measure of the mutual trust between them. We acknowledge that there can be two limitations of such an approximation. First, the absence of dialogue online between a pair of users does not imply a lack of trust (false negatives). Roughly 35% of the users on average do not engage in active conversation during the respective time period of the events. Second, conversation length could also reflect discord as opposed to agreement or trust (false positives). We manually scrutinized the crisis related Twitter data in detail and found no evidence of lengthy discordant conversations; contextually, in such situations, discord is unlikely to occur. The conversation lengths in the datasets range from 2 to 13.6 on average. We computed the Pearson correlation coefficient between the length of a conversation between a pair of users (i.e. the number of times they replied to each others tweets during the time period of the particular event), and the corresponding trust score obtained by our methodology. As Table 8.7 and the last two rows of Figure 8.2 show, moderate (0.45) to strong (0.85)

correlations were obtained between trust scores and conversation length, which indicates that this metric can be a suitable approximation despite the limitations noted above. We also note that conversation length was not used for computing any of the factors in our trust equation, it is an independent measure of quality.

Results: Figure 8.2 illustrates a series of surface heat plots varying two parameters: α and γ (β can be inferred from them). The first row of plots shows the variation in NDCG scores while the next two rows of plots show the variation in the correlation between pairwise trust score and conversation length, as the parametric values are varied, with and without valence. We find that the *inclusion of valence reduces the sensitivity of the method to parametric tuning in both cases*. Without valence, the NDCG scores range from 0.55-1 with high fluctuations on small parametric changes, whereas with valence the decision surface is more stable (the NDCG scores range from 0.75-1). We found the best NDCG score of 0.955 for CiaoDVD and 0.913 for Epinions (see Table 8.4 for parameter values). The surface plots for the social media datasets show similar trends: i) moderate (0.45) to strong (0.85) correlation between obtained trust values and conversation length demonstrating the effectiveness of our method; and ii) valence plays a critical role in a performance improvement and stabilizing the sensitivity of the method to parametric changes i.e. reducing fluctuation in correlation scores for small parametric changes. For the social media datasets, the best correlation values obtained by our method can be viewed in Table 8.7.

For each dataset, the best parametric setting is noted in Table 8.4, used henceforth for all experiments unless otherwise stated. Scanning through this table, it is clear that influence is the strongest factor contributing to an accurate ranking; α values typically range between 0.6 and 0.7. Content-based user similarity conditioned on topic popularity and

valence weighted by γ follows, with parameter values typically between 0.2 and 0.3. Finally, structural cohesion had a non-trivial but muted role with β values up to 0.2.

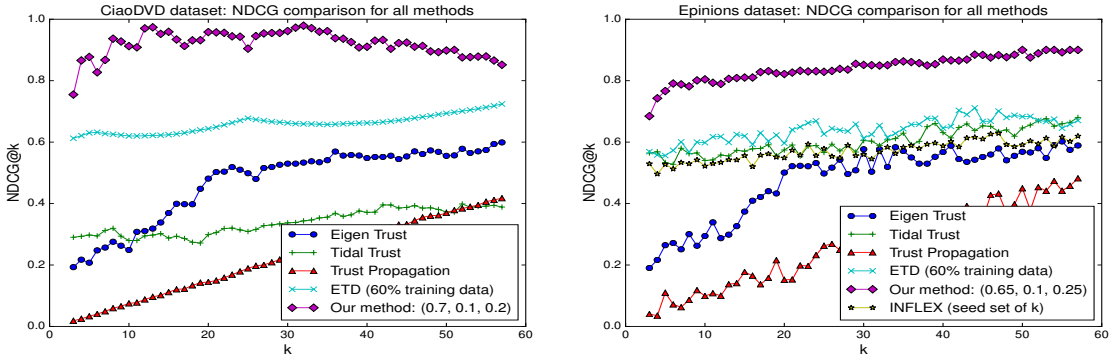


Figure 8.3: Comparison of NDCG score at various ranks for our algorithm against baselines, for CiaoDVD and Epinions.

8.5.3 Comparative Analysis

Baselines: We evaluated our method's performance on the CiaoDVD and Epinions datasets against the following unsupervised baseline trust computation algorithms: EigenTrust by Kamwar et al [147], TidalTrust by Katz et al [149] and an atomic trust propagation based approach by Guha et al [107]. We additionally include as baselines algorithms that only compute social influence to evaluate how they fare against trust computation; namely, a topic-aware influence maximization approach [8] called INFLEX, and a modification of our method to consider only the influence component i.e. setting the value of the parameter α to 1. Although our method is unsupervised, for the sake of completeness we further compare our method with a recently proposed supervised algorithm ETD [18], which also uses emotional information to aid in trust prediction, and is shown to outperform other

supervised trust prediction methods [127, 176]. While such algorithms are hard to deploy in a social media setting (as obtaining ground truth for trust relationships is difficult), and moreover expensive to compute, they provide a challenging baseline.

Results: In Figure 8.3, the x axis represents the rank at which we compute the NDCG score between the global ranking developed by the trust algorithm and the ground truth trust ranking, and the y axis represents the value of the NDCG score. Since we lacked the timestamp at which users posted and rated reviews for the CiaoDVD dataset, which is required for influence maximization algorithms, we could not run the INFLEX algorithm on this dataset. We observe that our method outperforms the baseline techniques, including the supervised method, giving a consistently high NDCG score > 0.75.

In order to assess the quality of pairwise user trust relations our algorithm is able to detect, we next compute the F1-score for the baseline algorithms and our method. We consider a reasonable trust relation to exists between two users if the value of the (normalized) pairwise trust between them is ≥ 0.5. We find from Tables 8.5 and 8.6 that our unsupervised framework *outperforms all baseline algorithms significantly with respect to NDCG scores at rank 50*, for both the social media and non-social media datasets. Adding content and valence information while computing trust does improve NDCG scores, showing that it is insufficient to consider only influence propagation while computing trust. Since EigenTrust, INFLEX and the influence component of our method (α set to 1) only compute a global and not pairwise trust score for each user, we don't compute an F1-score for them. Based on the available F1-scores, our method outperforms all unsupervised baselines and is comparable to the supervised ETD approach, when it trains on 60% of pairwise trust values (it outperforms the ETD approach with 30% training data). Clearly, it is impractical to obtain such a large fraction of ground truth pairwise trust data from even a moderately sized social network.

Algorithm	Dataset	NDCG@50	F1-score
EigenTrust	CiaoDVD	0.523	N/A
TidalTrust	CiaoDVD	0.35	0.43
TrustPropagation	CiaoDVD	0.33	0.21
ETD-T60 (T30)	CiaoDVD	0.69 (0.489)	0.78 (0.53)
Our method ($\alpha = 1$)	CiaoDVD	0.686	N/A
Our method	CiaoDVD	**0.955**	**0.84**
EigenTrust	Epinions	0.494	N/A
TidalTrust	Epinions	0.586	0.465
TrustPropagation	Epinions	0.371	0.237
ETD-T60 (T30)	Epinions	0.645 (0.51)	**0.816** (0.52)
INFLEX	Epinions	0.579	N/A
Our method ($\alpha = 1$)	Epinions	0.658	N/A
Our method	Epinions	**0.913**	0.805

Table 8.5: NDCG at rank 50 and F1-score values for different algorithms for CiaoDVD and Epinions. ETD results are reported with 30% (T30) and 60% training (T60) data.

Algorithm	Avg NDCG@50 for Twitter datasets	Avg F1-score for Twitter datasets
EigenTrust	0.444	N/A
TidalTrust	0.549	0.511
TrustPropagation	0.321	0.303
INFLEX	0.647	N/A
Our method	**0.876**	**0.803**

Table 8.6: NDCG scores at rank 50 and F1-scores averaged across the Twitter datasets, for the unsupervised algorithms using conversation length as ground truth.

For social media datasets, in Table 8.7 we report the correlation between conversation length and pairwise trust scores computed by our method and the unsupervised baselines (supervised baselines are not feasible because we lack pairwise trust ground truth). As before, we observe that our method outperforms the baselines significantly.

Dataset	Corr (Tidal Trust)	Corr (Trust Propagation)	Corr (Our Method)
India Anti-Corruption	0.35	0.16	**0.507**
Mumbai Blast	0.521	0.187	**0.849**
Phone/Tablet	0.309	0.083	**0.4614**
Houston Flood	0.267	0.111	**0.484**
Hurricane Irene	0.222	0.091	**0.518**
Nice Attacks	0.313	0.181	**0.565**

Table 8.7: Correlation between pairwise trust score and conversation length between pairs of users during that time period, using different algorithms

8.5.4 Performance Enhancements and Scalability

Benefits of Degree-Discounting: Figure 8.1 examines the benefits of accounting for degree-discounting while computing pairwise user trust. The user trust relations obtained without this optimization achieve an NDCG@50 score of 0.76 and an F1-score of 0.424 on the CiaoDVD dataset. Adding degree discounting without valence yields an NDCG@50 score of 0.82 and F1-score of 0.63. Further, adding valence yields an NDCG@50 score of 0.955 and an F1-score of 0.84. Similar benefits were observed in other datasets.

Execution Time: We present an analysis of the total runtime of our method amortized over the number of user pairs in Table 8.8. We note that using SentiStrength for detecting valence (which can process about 10K tweets per second), the efficiency of computing pairwise trust is comfortably able to handle the influx of tweets associated with an event even at full firehose (several hundred million tweets a day) rates.

8.5.5 Case Study: Crisis Response

Here, we examine the kinds of individual users or organizations on Twitter who were identified by our algorithm as being highly trustworthy during various hazards, and also track the change in their overall trustworthiness across different temporal phases of the

Dataset	Runtime per pair of users
India Anti-Corruption	0.000854 sec
Mumbai Blast	0.0034 sec
Phone and Tablet	6.56×10^{-4} sec
Houston Floods	0.000296 sec
Hurricane Irene	6.689×10^{-5} sec
Nice Attacks	4.659×10^{-6} sec
CiaoDVD Film	0.000199 sec
Epinions	9.616×10^{-6} sec

Table 8.8: Runtime for pairwise trust value computation and parameter tuning, amortized over number of user pairs

hazard occurrence, i.e. before the hazard, during its worst impact and immediately after it has subsided at the affected area. Due to space constraints, we only display results for three hazard datasets in Table 8.9. Users common across columns are represented in the same color (except black) so that we can track how their trust ranking changes across the three phases of the hazard. Top trustworthy users unique to each phase are colored in black.

Observations: Several users in Table 8.9 are well known personalities or reputed organizations and are thus likely to be highly trusted. They include journalists and news agencies (*cnnbrk, BreakingNews, Breaking911, ndtv*), key weather officials and services (*NWSHouston, CraigAtFEMA, JimCantore*), entertainment related (*mashable, funnyordie, GlitchxCity*), political and social figures (*BarackObama, xanpearson, dina*) and relief organizations (*SamaritansPurse, RedCross, HumaneSociety*). This lends credibility to our analysis.

Moreover, certain users were not previously well known in any field, yet our algorithm identified them as commanding a high trust value during the time period of these events. Such users have been highlighted in italics in Table 8.9. On examining their tweet logs, we found that the Twitter user *TexasTsunami* provided assistance to the Houston flood victims

Before	During	After
mashable	cnnbrk	cnnbrk
lfmccullough	BreakingNews	BreakingNews
richmintz	USArmy	BarackObama
Reuters	CharityIdeas	nytimes
FrommersTravel	severestudios	HumaneSociety
nydailynews	shibanijoshi	RedCross
6abc	JimCantore	JimCantore
travelingmoms	xanpearson	*atmanes*
severestudios	CraigatFEMA	CraigatFEMA
BreakingNews	MikeBloomberg	*Fanua*
Daily_Press	*atmanes*	RedCrossPhilly
cnnbrk	*Fanua*	SamaritansPurse
funnyordie	HumaneSociety	RedCrossSAZ
afreedma	BarackObama	mashable
BarackObama	RedCross	USGS

(a) Hurricane Irene

Before	During	After
GlitchxCity	Breaking911	Breaking911
Sportsnaut	NWSHouston	NWSHouston
TriCityHerald	AlertHouston	BarackObama
Nick_Anderson_	WSOCWeather	AlertHouston
HOUBizJournal	StormViewLIVE	*TexasTsunami*
StarfishGawdess	BarackObama	RedCross
Breaking911	ArchCollegeTAMU	ArchCollegeTAMU
BarackObama	JohnCornyn	WSOCWeather
JohnCornyn	*TexasTsunami*	JohnCornyn
NWSHouston	GlitchxCity	GlitchxCity

(b) Houston Floods

Before	During	After
ScepticGeek	ndtv	mid_day
AltCricket	mid_day	ndtv
acarvin	rameshsrivats	htTweets
dina	maheshmurthy	rameshsrivats
maheshmurthy	AnupamPKher	*ashwinsid*
htTweets	htTweets	Netra
mid_day	fakingnews	fakingnews
ndtv	PatrickMeier	maheshmurthy
timesofindia	*ashwinsid*	PatrickMeier
guardian	Netra	ScepticGeek

(c) Mumbai Blast

Table 8.9: Top trustworthy users with global trust score$>$ 0.7 for different datasets. The same color is used for users reappearing across the phases of a disaster. Newly emergent trustworthy users are italicised.

and tweeted some strong statements such as "*I know it is ILLEGAL to give a hungry man a sandwich here in Houston. Is it legal to assist flood victims?*", which popularized him and increased his trust level on topics related to the flood (though he wasn't as popular as the other highly trusted users). Similarly, user *atmanes* helped a taxi service in transporting animals, while user *Fanua* repeatedly re-tweeted several reliable news sources regarding Hurricane Irene, which increased the trust of other users towards them during this time. User *ashwinsid* provided transport to stranded victims of the Mumbai blast. Thus, *our approach not only captures trusted users who are well known or popular but also effectively identifies emergent trustworthy users* who were neither well known nor previously popular – during and immediately after an emergency situation. This is particularly relevant for our ongoing efforts in identifying trustworthy "good samaritan" citizen sensors both during and immediately after a disaster occurs.

Role of Valence: Valence is crucial in recognizing emergent trustworthy users. Without valence, we observed that good samaritan, trustworthy users *(atmanes, Fanua, TexasTsunami, ashwinsid)* could only be detected in one of the datasets (Mumbai Blast), and that too only in the top 60 trustworthy users (and not in the top 10 or 15).

Dynamics of Trust: Finally, we perceive intuitive trends in connection with the changes in highly trusted users over the three phases of the disaster previously outlined. From Table 8.9 we note that before the hurricane or flood struck the respective affected area, the highly trusted users were from a wide variety of fields such as travel, sports or entertainment blogs along with key political figures and news channels. However, we see a significant variation during and immediately after the disaster. News channels, reporters and weather related user accounts rise to the top of the list. New trustworthy users start emerging in the 'during' phase and their trust scores rise as the disaster progresses to the 'immediately after' phase,

which is when they primarily provide voluntary assistance to affected victims. Additionally, relief and humanitarian organizations such as the *RedCross* are highly trustworthy shortly after the disaster.

8.6 Conclusion

In this chapter, we presented a simple unsupervised approach to computing inter-personal trust among users within a social network, bringing to bear theoretical ideas from psychology and sociology as they relate to influence, passive (structural) and active cohesiveness (content and valence). In our empirical analysis we sought to answer three questions: i) Which among the three factors (influence, structural cohesion, affective valence) are most important to estimate trust relationships among users in a social network? ii) How robust is the method in predicting trust with and without valence? iii) How does real-world performance of the method compare to baselines and can it be useful in emergent settings?

Consistent with the literature, although we find influence to be the principal factor contributing to recognizing trustworthy users in social media, we demonstrated that the presence of valence or sentiment while calculating trust adds significantly to the stability of the response surface, and to performance. Structural cohesion, while still useful has less of an impact than the other two factors, especially in the case of ephemeral (Mumbai attacks, Nice attacks, Hurricane Irene and Houston Floods) and political movement events (India Anti-Corruption). This may be related to the type of event, because structural cohesion plays a stronger role than valence-enhanced content on the Phone and Tablet dataset. Finally, we demonstrated that our method is able to develop a trust based ranking of users that comprehensively outperforms strong baselines on a range of real-world datasets. We also

demonstrated the efficacy of the methodology for the dynamic identification of trusted users in emergent crisis settings.

We would like to pursue the following directions as part of future work. First, while using valence or sentiment as part of our trust computation approach, we would like to make a subtle distinction between 'evaluative' valence and 'affective' or 'emotional' valence. Evaluative valence identifies the extent to which respective language expressions have a cognitive or rational effect, as opposed to an affective or emotional valence [263, 264]. For example, two users might use the words *terrific* and *love* respectively while expressing their views on a certain topic. Though both users convey an overall positive valence on the same topic, the word 'terrific' denotes a more practical connotation than the word 'love', which is more emotional or extreme. This nuanced distinction can impact the extent to which this pair of users trust each other, and is worth investigating.

Second, since we have found social influence to be an important component of predicting dyadic trust in social networks, we would like to study if the reverse is also true, i.e. can our technique of understanding user trust be utilized to analyze the future social influence of users among their network? This can unearth the correlation between social influence and social trust in greater detail.

Chapter 9: Emotional and Linguistic Online Behavior Patterns Focused on Clinical Depression

In this Chapter, we continue our exploration of user behavioral characteristics and their interactions with other users to tackle the pragmatics problem. Guided by appropriate social and psychological research, we conduct an observational study to understand the interactions between clinically depressed users and their ego-network when contrasted with a differential control group of normal users and their ego-network. Specifically, we examine if one can identify relevant linguistic and emotional signals from social media exchanges to detect symptomatic cues of depression. Based on our observations, we then describe an approach to extract relevant features and show that building a classifier to predict depression based on such features can achieve an F-score of 90%.

9.1 Introduction

The health and developmental outcomes in modern society are often shaped by peer interactions. Relationships with family and care-givers during childhood and additionally peers from adolescence to adulthood, are critical to understanding both dimensions of well being and sources of risk during different phases of life [332]. An increasing amount of such interaction happens online via various social media platforms such as *Facebook* and *Twitter*. In fact as noted by a recent report from the American Academy of Pediatricians (AAP) [233]

and echoed by a recent Pew study [249], social media interactions now represent a key communication modality for the vast majority of US adolescents and young adults and a significant fraction of older adults.

Given the pervasive use of social media and evidence presented by such studies and reports, a key question then to ask is whether such use departs significantly from those found in physical (offline) social networks, as studied by sociologists and psychologists for many decades. Specifically, our goal in this observational study is to study such effects on a subset of the US population that is active on Twitter, paying particular attention to interactions between clinically depressed individuals and their ego-network. We also seek to examine if one can identify relevant signals from social media activity, engagement and linguistic content to detect symptomatic cues of depression. We believe that such studies build on and can deepen our understanding of previous efforts [59, 64], and represent an important step towards understanding the impact of social media use on mental health and well-being.

To facilitate the analysis of our observational study, we examine network effects related to participation, engagement and ego-neighborhood. We define network participation features to include both passive (tweets a user is exposed to, retweets or mentions a user receives) and active participation (mentions, retweets and conversations made by the user). We define network experience features to include both content (e.g., linguistic cues, emotion) and relational dynamics (e.g., conflict/support, influence) of network embedded interactions. We also examine neighborhood effects and analyze key statistics of the neighborhood such as size, centrality and affinity to form clusters or communities. Our study includes both depressed users and their ego-net(s) as well as normal users (control) and their ego-net(s). Some of our key findings include:

- With respect to participatory statistics, users suffering from depression tend to: i) post less frequently as well as later in the evening when compared to their normal counterparts, which agrees with offline studies; ii) have smaller networks with densely clustered pockets; iii) less frequently refer explicitly to their network partners online via retweets and mentions; and iv) have a slightly lower regional entropy for their ego-network (i.e. higher co-location within ego-net) as compared to the non-depressed class of users (see Table 9.1).

- With respect to engagement and responsiveness, the results largely agree with offline studies, that users suffering from depression are less engaged with their network, and neither influence nor are influenced by their network to any significant extent (see Table 9.2). However, we do find depressed users receiving reasonable social capital from their online neighbors in terms of reacting to their state of mind via supportive tweets (see Figure 9.1).

- There is a strong presence of linguistic cues such as self-focused pronoun usage by depressed users online, supporting various offline studies. The differential analysis with respect to the normal class of users is particularly compelling (see Figure 9.2).

- A majority of depressed users exhibit strong negative emotion in their tweets while their general ego-network tends to be positive (see Figure 9.5). Moreover, for most depressed users we do not observe much periodicity in their emotional signal with respect to days of the week. On the other hand, non-depressed users tend to be more positive and correlated with their ego-network, showing typical trends of being less positively valenced in their posts during the start of the work week and more positive towards the weekend, agreeing with psychological theory (see Figure 9.4).

The cues above allow us to build a feature set comprising network, content and user based features which illustrates that in an unsupervised setting, normal users and depressed users are separable. In a supervised setting, we find that using the same feature set one can build a classifier to classify depressed individuals, which achieves an F1-score of 90%.

9.2 Related Work

Several experiments have been conducted and theories posited in the fields of social science, psychology, psychiatry, medical science and linguistics in conjunction with the onset and spread of clinical depression and its symptoms in individuals [31, 124, 133, 141, 150, 164, 197, 219, 232, 270]. While the importance and utility of such empirical research cannot be underestimated, a key challenge associated with it is the difficulty in obtaining data pertaining to specific individuals, as well as monitoring them for long periods of time. Therefore in recent years, researchers have been employing social networking websites in order to collect data as well as study behavioral characteristics of people related to various aspects of mental and psychological health. Social media has been used to study dissemination of health information [117, 278], as well as to gain key insights related to the spread of diseases and their symptoms [60, 240]. Prior work [45, 59, 60, 64–66] has also highlighted the usefulness of social media in various issues concerning mental health. Jelenchick et al [136] and Moreno et al [219] analyzed the phenomenon of escalating signs of 'Facebook Depression' among users due to rising use of the social network Facebook. Changes in mood and emotional state of individuals is reflected on their social media profiles, according to multiple studies conducted on Twitter data [25, 99]. Park et al [239] observed that people make posts online regarding their depression and even treatment received. Analyzing textual content of individuals has also proved to be helpful in identifying signs of

various mental disorders among them [31, 133, 227, 234, 245, 270, 361]. DeChoudhury et al in [64] use social media as a tool to study postpartum depression in pregnant women. In another work [65], they leverage social media analysis to estimate an individual's risk of having Major Depressive Disorder (MDD). Our work builds on these efforts, drilling down on emotional, linguistic (self-focused pronouns) and location-based cues in addition to standard activity patterns and features of the ego-network. We study emotional contagion and temporal relationships between depressed individuals and their ego network, as well as their orientation in their network topology. Furthermore, we show how to realize a practical and accurate classifier for potentially classifying users who may be suffering from depressive tendencies by focusing on seven high level features.

9.3 Data Collection

Before describing our methodology and key questions we seek to study, we discuss our data collection methodology. We emphasize here that our study is observational with no direct engagement with Twitter users [15]. To identify candidate depressed Twitter users for our study, we first collected a set of terms commonly used in conjunction with the word 'depression' from the depression glossary at *www.webmd.com*. We then crawled the Twitter streaming API to extract a sample of tweets mentioning any of these terms, only retaining tweets from regions in the US. After filtering out user accounts providing depression-related medical help, we identified candidate users mentioning such terms frequently. Within this subset we then focused on users who explicitly reported being on anti-depression medication; the names of pharmaceutical drugs typically used to treat clinical depression in the US were

[15]OSU's Office of Responsible Research has determined that this study does not meet the US federal definition of human subjects research requiring review and neither IRB review nor exemption review is required. This determination is issued under The Ohio State University's OHRP Federal wide Assurance #00006378.

obtained from a collaborator. Fifty such users spread across the US were then identified as our ground-truth depressed user class.

We next used a Twitter Streaming API-based crawler to collect seven months worth of Twitter data, from July 2016 to January 2017, consisting of the tweets of the above identified clinically depressed users, their immediate or one-hop neighbors (a user's followers and followees i.e. friends on Twitter) and their two-hop neighbors (the followers and followees of a user's followers and followees on Twitter). We also used this dataset to expand the depression lexicon we constructed from *www.webmd.com*, to include social media specific terms. For this, we trained a word2vec [213] model on the Twitter dataset, extracted from it the top 500 words most likely to be used in the context of the depression-related set of terms, and added these to the depression lexicon. Further details of the Twitter dataset are presented in Table 9.1.

For the control group of 'normal' (non-depressed) users, we elected to randomly sample a group of a hundred users based in the US. We explicitly sought to minimize any network interference effects with any of our selected depressed users (i.e. the ego-net of normal users we sampled had negligible overlap with depressed users' ego-nets), by discarding users who did not meet this criterion. The overlap exceptions being highly popular users (such as rock stars, famous sports personalities, major league teams etc.) who may appear on both depressed users' and normal users' ego-nets.

We note that Table 9.1 shows some interesting differential statistics between normal and depressed users in terms of the size of respective ego-nets and activity levels. We will drill down on some of these aspects in the subsequent sections.

Table 9.1: Participatory Statistics of Users' Ego Network. Measure values are averaged over all users for both the depressed and normal classes, except for the median time of posting.

Type	Measure	Depressed	Normal
Activity	No. of posts (daily)	5.84	7.95
	No. of posts (entire period)	2041.56	3145.88
	Retweet rate (daily)	4.61	7.28
	Retweet rate (entire period)	1366.54	2742.32
	Mention rate (daily)	1.68	4.25
	Mention rate (entire period)	359.78	1048.45
	Median time of posting	11:51 p.m	5:36 p.m.
	Regional entropy of ego-net	3.761	4.483
Specific ego-net properties	Size (1-hop)	1196	3215
	Size (2-hop)	210850	987098
	Density (1-hop)	8.59×10^{-5}	2.67×10^{-5}
	Density (2-hop)	3.44×10^{-7}	1.41×10^{-7}
	User clustering coeff (1-hop)	0.208	0.073
	Eccentricity of user (1-hop)	4.4	2.6

9.4 Methodology

Informally, we seek to build a model to predict which users are likely to become victims of clinical depression in the near future based on their behavioral characteristics on social media, and to also identify highly depressed users within a social network. We would additionally like to answer the following question: Is the emotion of a depressed user on a social network a function of what a user is exposed to (i.e. via a user's immediate or one-hop neighborhood), and a secondary function of what the user's neighborhood is exposed to (i.e. the user's two-hop neighborhood)? We focus on *Twitter* as the social network of choice.

9.4.1 Network Activity and Participation

Theory: Kawachi et al [150] have shown that depressed users tend to cluster together. Lustberg et al [197] found a strong correlation between depression and insomnia, i.e. depressed users tend to be more active online during the later hours of the night. Also, studies

both offline among college students [164] as well as on social networks (Facebook) [219] have shown an increase in social media/internet usage in victims of depression.

Analysis: Table 9.1 provides basic statistics regarding the online activity of users of both classes on Twitter, as well as various structural features of their ego-network. We find that in line with social and psychological research, the potentially depressed users typically exhibit more of nocturnal behavior than non-depressed class users (see median time of posting). Such users tend to mention and retweet other users in their posts less frequently than their non-depressed counterparts, suggesting a lack of direct interaction with other users. Though the overall activity of the depressed class users is lesser than a non-depressed class user, a smaller percentage of the posts tweeted by depressed users consist of simply retweeting what others have said, as compared to normal users who retweet others much more.

In addition, we computed the regional entropy of the depressed users' ego-network, using the algorithm by Compton et al [58] to extract the location of each tweet. Interestingly, depressed class users on Twitter have a slightly lower tweet location entropy than the non-depressed class of users, suggesting that in terms of their physical geo-location, people in a depressed user's ego-net are closer to each other than those in a normal user's ego-net.

With respect to ego network characteristics, we observe that the one-hop and two-hop networks of depressed class users are smaller on average when compared to normal users. Their ego networks tend to consist of multiple, tighter-knit clusters (hence the higher density of the network), and the depressed users themselves seem to be connected with a smaller fraction of nodes within their ego network than the normal users. The eccentricity of a depressed user is also significantly larger than that of a normal user within their respective one-hop ego-networks, suggesting that normal users tend to be more centrally located within the ego-network. A somewhat surprising statistic is that in spite of having a lower

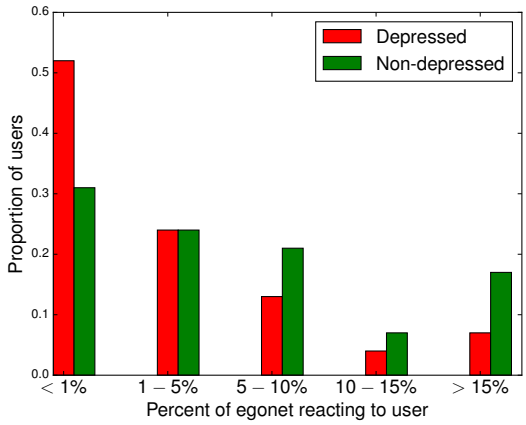


Figure 9.1: Percentage of ego-network reacting to a user, in the form of mentions, retweets or replies, for both user classes.

eccentricity value, the average clustering coefficient of depressed users is quite a bit higher. The *raison d'etre*, as we shall demonstrate shortly, is because of the homophily effect [267] – depressed users tend to be clustered with other potentially depressed users within their ego-net (see Figure 9.3).

9.4.2 Network Engagement and Experience

9.4.2.1 Network Responsiveness to User

Theory: Leading sociologists and psychologists note that victims of clinical depression tend to be socially isolated in an offline setting. In a study, Joiner et al [141] tested whether depressed individuals would be affected by their non-depressed peers in one-on-one interactions, and they found that victims of depression often receive from their peers a negative or unfavorable response or rejection to their constant seeking of reassurance, which in turn exacerbates their depressed state of mind. The fact that depressed persons prefer to associate with others who also tend toward depression (homophily) was also concluded by

Rosenblatt et al [267] through an experiment with undergraduate students. They found that depressed people felt worse than earlier after speaking with non-depressed people, but not after speaking with similarly depressed targets. In yet another study on adolescents, Hogue et al [124] conclude that as a result of the 'selection effect', they tend to choose friends possessing similar levels of internal distress.

Analysis: In order to examine whether the above ideas extend to online social networks, we study basic network engagement effects of the depressed and non-depressed users in our study. Figure 9.1 displays the percentage of the users' ego-network reacting to the user in the form of mentions, retweets or replies. This gives an indication of the extent to which a depressed class user is able to influence his ego-network. We observe that nearly half of the depressed users have no or minimal impact on their network (network response towards them is $< 1\%$), while for roughly the other half, 5-15% of their network reacts to them in the form of replying to them, mentioning them or retweeting their tweets. This implies that the depressed user in these cases is engaged and able to exert some influence on his ego-network. For normal users the level of engagement is distributed between 0-15%. Certain depressed users receive a significant amount of support from their ego network ($> 15\%$). The results here suggest that in the online setting while some users tend to be socially isolated (agreeing with some of the above offline studies), some of them are adequately engaged with their ego-networks both in terms of participation (Table 9.1) and engagement (Figure 9.1).

We next examine the relative position of a user with respect to his/her network structure. We aggregate all the tweets made by the depressed users and the users belonging to their ego-network, and obtain vector representations of each word in the tweets from our pre-trained word2vec model (described in Section 8.5.1). For the purpose of this evaluation, only original content posted by a user or ego-net partner is levered; retweets are not considered

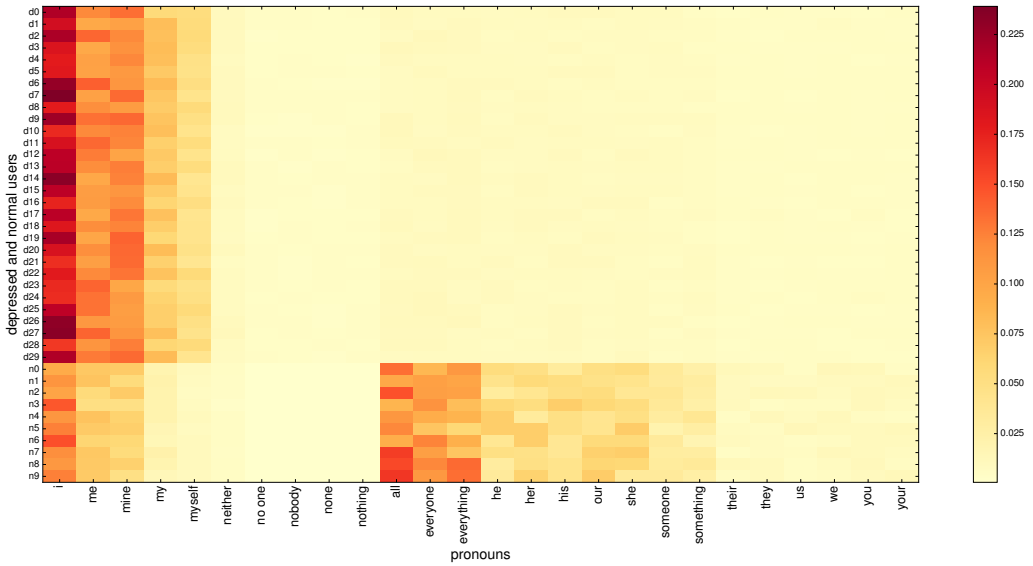


Figure 9.2: Heatmap distinguishing linguistic pronoun usage of depressed users from normal users. Self-focused (e.g. 'I', 'me', 'my', 'mine') and group connotation pronouns (e.g. 'our', 'we') have the highest differential capability between the two classes. In the interests of space, we present a representative sample of non-depressed and depressed users here.

in our aggregation to avoid bias effects. After accumulating the vector representation of each word for each user, we perform a dimension-wise average of all the vectors to get a single high-dimensional vector representation for each user. To visualize this, we use Multi-dimensional Scaling (MDS) [203] with cosine similarity as a similarity metric, to scale down the users' feature vectors to two dimensions. We depict this in Figure 9.3 for a subset of users belonging to both classes. The x and y axes represent the two dimensions obtained from MDS. For each user plot, the green points represent the users belonging to their ego-network while the large red point represents the users themselves. We observe that the depressed users are like outliers in their networks, at a significant distance away from the core of their network. Multiple potentially depressed users tend to cluster together (the pink points – more on this in Section 9.5). The non-depressed class users in the last row of Figure 9.3 are more centrally located and are similar to the other users in their network.

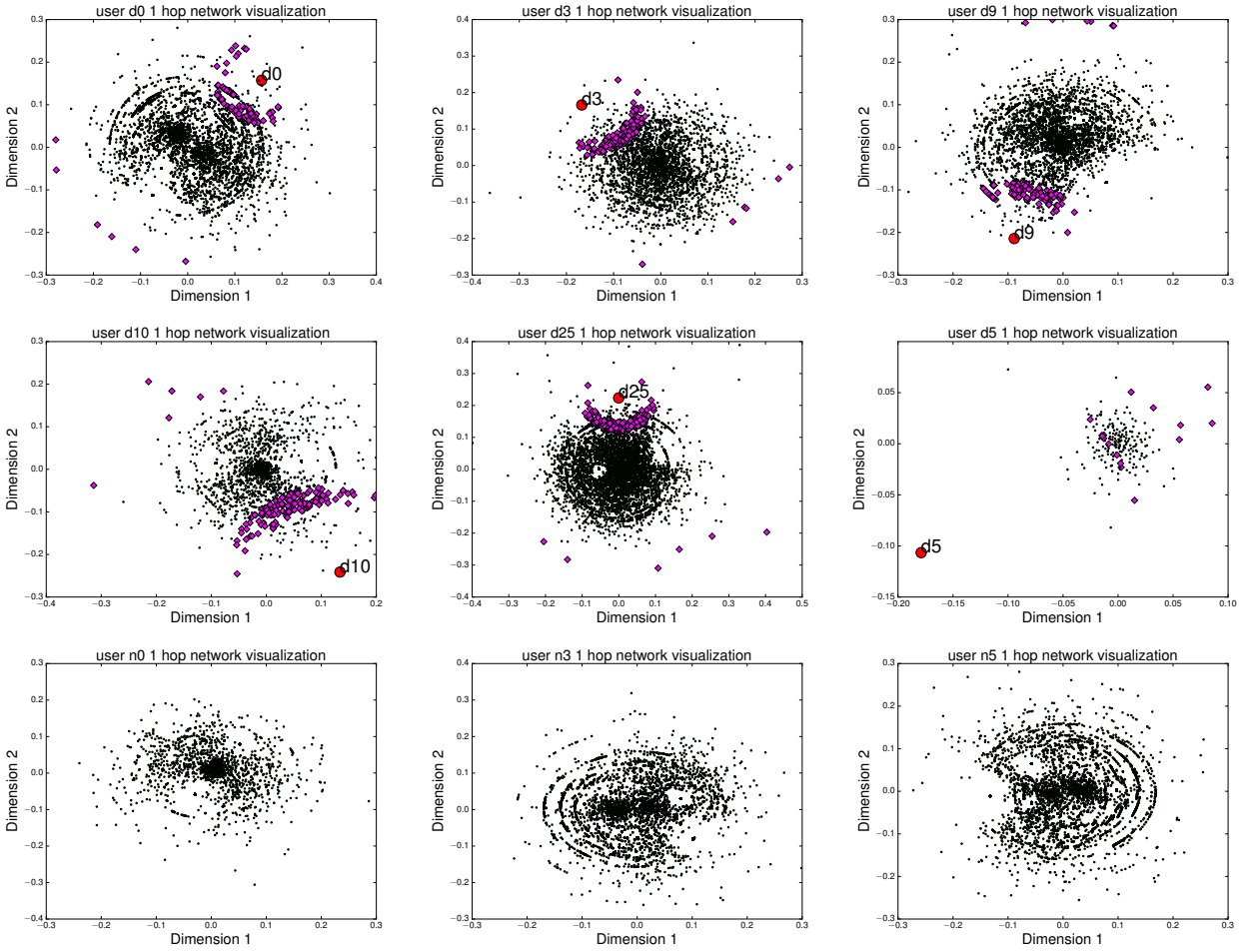


Figure 9.3: Visualization of selected users belonging to the depressed and normal class within their ego-net. The x and y axes represent the 2 dimensions obtained from multi-dimensional scaling. The green points represent the ego-net while the red point represents the user. The pink points represent the users in the ego-net of the depressed user, who have also been predicted as depressed. The first two rows of plots belong to depressed users and the third row belongs to the normal class of users.

9.4.2.2 Linguistic Content (Pronoun) Analysis

Theory: Various studies have analyzed the linguistic style and content associated with the text and/or speech of depressed individuals [31, 133, 270]. These attest to the fact that depressed individuals have a higher propensity towards self-focus, which translates to an increased linguistic usage of personal pronouns associated with the self such as 'I' and its derivatives, and a reduced use of third-person pronouns or those exhibiting collective connotation.

Analysis: In connection with the above findings, we explore the linguistic style patterns of the depressed and non-depressed class users in terms of the pronoun usage in their tweets. We first identify the top pronouns that are most frequently used by the users of both classes in their tweets. Figure 9.2 shows a heat map in which the colorbar represents the frequency of the pronouns with respect to the total number of unique words in the tweet vocabulary. In this figure and subsequently, the usernames beginning with the letter 'n' represent normal class users and those beginning with the letter 'd' represent depressed class users. We observe perfect separation of the users into two classes. We observe that the self and negative connotation pronouns (the first ten pronouns on the x-axis) are used relatively heavily by the users belonging to the depressed class, and second-person pronouns such as 'you' or those that denote group connotations such as 'we', 'our', 'they' etc are hardly used by this class of individuals. This indicates that these users are more inclined to talking about themselves in an isolated manner, without including themselves with other people.

9.4.2.3 Content-based Emotion Analysis

Theory: Multiple analyses related to the spread of depressive symptoms were performed on a densely interconnected social network of 12067 people as part of the Framingham Heart

Study [85]. Depressive symptoms were evaluated using CES-D (Center for Epidemiological Studies Depression Scale) scores, and results confirmed that both low and high CES-D scores (i.e. absence and presence of depression in an individual) in a given period were strongly correlated with the CES-D scores in the individual's friends and neighbors, extending up to three degrees of separation (one's friends' friends' friends i.e. the three-hop neighborhood of an individual). Interestingly, this study also confirmed that while positive emotions such as happiness seem to spread across a social network (conditioned on geographical location), negative emotions such as sorrow, anxiety, distress or depression do not possess the same contagion effect i.e. they do not spread across a social network. Studies in the literature have also analyzed the affective valence associated with the text and/or speech of depressed individuals, and have found the presence of negative emotional affect to a much higher degree in the language of victims of clinical depression [133, 232].

Analysis: We first aggregate all the tweets that have been collected for each user and their ego-network on a per-day basis. For the purpose of quantifying the emotional strength within the textual content expressed by users in their tweets, we use a tool called SentiStrength [323] that is customized to detect positive or negative emotion within short, informal texts characteristic to social media.

Figure 9.4 displays the distribution of average positive and negative emotion over the seven days of the week, for selected users belonging to the depressed and non-depressed classes. The tweets of the user as well as their network were first aggregated according to the day of the week on which they were posted and then segregated into positive and negative after averaging the predominant emotion expressed in their content. The x-axis represents the day of the week and the y-axis shows the number of tweets of the user or their network expressing positive or negative emotion. We can see from the stacked bar chart that for users

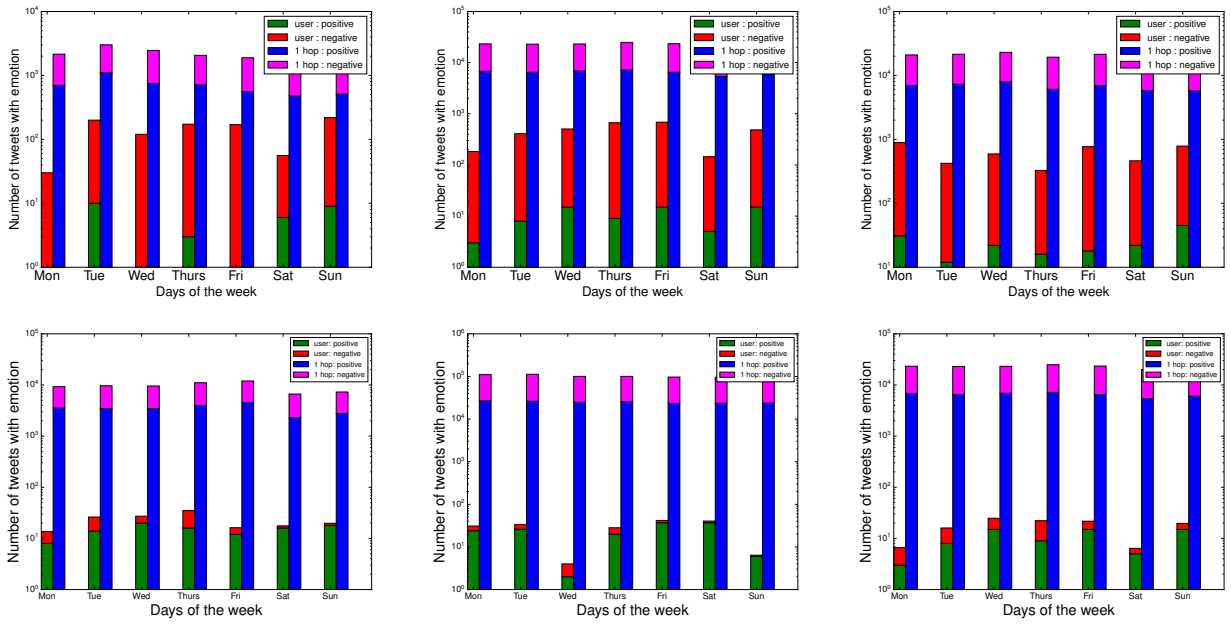


Figure 9.4: Number of tweets with positive and negative emotions of users and their one-hop and two-hop networks, over days of the week. The first row of figures is for the depressed class of users and the second is for the non-depressed class.

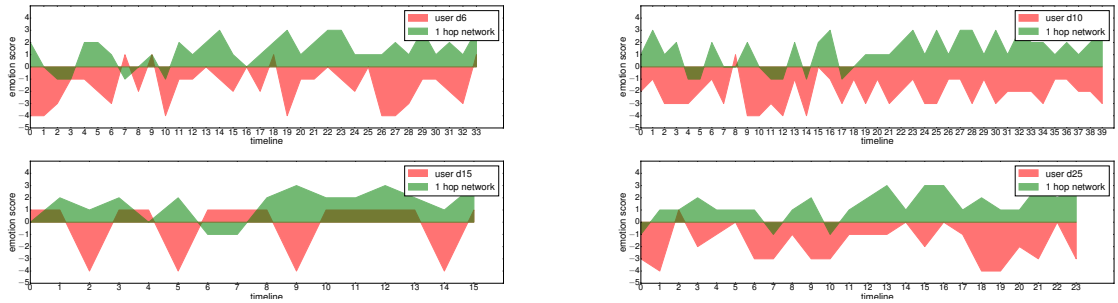


Figure 9.5: Emotion scores of selected depressed class users and their one-hop networks over time (red represents the user's emotion and green represents the average emotion of the one-hop network). The brown regions show overlap between the emotion of depressed users and their network. One unit of time on the x-axis corresponds to three days of user activity.

Table 9.2: Cross correlation analysis of a selected representative sample of depressed class and non-depressed class users' emotion distribution over time with the users of their one-hop and two-hop network.

User[neg,pos]	1-hop[neg,pos]	Cor 1-hop (Lag/Lead)	Time 1-hop (Lag/Lead)	Cor 2-hop (Lag/Lead)	Time 2-hop (Lag/Lead)
d1 [-2, 1]	[-1, 3.5]	0.118	1	0.106	1
d5 [-3, 1]	[-1, 3]	0.209	-1	0.061	-1
d8 [-3, 1]	[-2, 3]	0.23	1	0.065	-1
d10 [-4, 1]	[-1.5, 3]	0.282	1	0.132	-1
d12 [-2, 1]	[-1.5, 3.5]	0.116	1	0.11	0
d14 [-2, 1]	[-1, 3.5]	0.078	-1	0.06	1
d15 [-3, 1]	[-2, 3]	0.213	1	0.128	0
d16 [-2, 1]	[-1.5, 3.5]	0.036	1	0.012	0
d21 [-2, 2]	[-1, 4]	0.017	1	0.031	-1
d22 [-3, 2]	[-1.5, 4]	0.025	-1	0.019	1
d24 [-2, 1]	[-2, 3.5]	0.186	1	0.177	-1
d25 [-3.5, 1]	[-2, 3]	0.277	1	0.218	-1
d29 [-4, 1]	[-1.5, 3]	0.301	1	0.237	-1
n0 [-2, 3]	[-1, 5]	0.516	-1	0.467	-1
n1 [-1, 4]	[-1, 5]	0.628	-1	0.818	-1
n3 [-1, 3]	[-2, 4.5]	0.65	-1	0.868	-1
n5 [-2, 3]	[-1, 4]	0.51	0	0.579	-1
n6 [-1.5, 3]	[-1, 4]	0.78	-1	0.76	-1
n8 [-1, 3]	[-1, 3.5]	0.517	0	0.511	-1

belonging to the depression class, the tweets expressing negative emotion heavily outnumber the tweets expressing positive emotion. We do not observe any significant relationship between the predominant emotion of a user and the day of the week, which largely agrees with social and psychological research. On the other hand for individuals belonging to the normal class, positive emotion is dominant and these users seem to exhibit more negative emotion in their tweets on the initial working days of the week i.e. Mondays and Tuesdays, and as they reach the end of the week their tweets grow more positive. The average emotion expressed by users belonging to the one-hop network as well as the two-hop network (not shown in the figure for the sake of brevity) of both classes is largely positive.

We next analyze the temporal distribution of the overall emotion expressed by the potentially depressed users and their ego-network (Figure 9.5). For this, we aggregate all the tweets made by a given user as well as their ego-network over three-day intervals, not considering the days when the user does not tweet anything. As earlier, we eliminate from consideration the retweets of a user. The x-axis represents the timeline, where one unit on the x-axis corresponds to three days, and the y-axis represents the average emotion score for that duration. The red and green plots represent the emotion distribution of the user and his ego network respectively. As expected, the emotion expressed by the depressed users is predominantly negative with some regions of positivity, whereas the overall emotion of the users' ego-networks is positive. The overlap in emotion of the depressed users with their ego-net over time (the brownish colored portions in the plots) is low. This confirms that a depressed user is unlikely to get influenced by the emotion prevalent in their neighborhood, and tends to remain socially isolated, in line with social and psychological studies.

We further strengthen our claim of the depressed sentiment of a user not affecting or being affected by his/her neighborhood to a significant extent. For this, we inspect the

cross correlation between the temporal emotion distribution of the users belonging to the depressed and non-depressed classes with that of their one-hop and two-hop networks (see Table 9.2). In order to compute this, as earlier, we aggregated the daily tweets of each user and their network, eliminated retweets, and the days when the user did not post any tweets. We then computed the cross correlation values over time with respect to the daily average emotion scores between the user and those of his one-hop network (third column of Table 9.2), and the user and those of his two-hop network (fifth column of Table 9.2). We investigated if we could observe a temporal lag (indicating that emotion permeates from the user's ego-network to the user) or lead (an emotion contagion from the user to his ego-network) associated with the highest cross correlation value (fourth and sixth columns of Table 9.2). A positive value in the fourth and sixth columns represents a lead while a negative value represents a lag. The parentheses in the first two columns contain the range of the emotion score for the users and their ego-net respectively, averaged over all their tweets.

The best correlation values of the depressed users' aggregated average emotion with that of their one-hop or two-hop network are quite low (≤ 0.3), while they are much higher for the normal class users (> 0.5 in most cases). We observe that for users belonging to the depressed class, the average emotion scores range between -2 and -4, while for the ego-networks as well as normal users, they are largely positive with slight negativity. Some normal class users such as $n0$ and $n5$ are slightly more negative than others. We do not find any significant evidence of a lag or lead in time between the emotion of a user belonging to the depressed class and his ego-network, indicating that the depressed users seem to be largely isolated from and unaffected by their neighbors and/or network. Normal users predominantly tend to lag behind their network i.e. appear to be influenced by the emotion of their immediate neighbors (average correlation value of > 0.5) within a day.

Table 9.3: Two-sample t-test of significance comparing both user classes (significant differences in bold)

Feature	**p-value**
Average user emotion	**0.000142**
Clustering coefficient	**0.05**
Pronoun usage	**0.004**
User Activity	0.112
No. of mentions	**0.00243**
No. of retweets	**0.00121**
Location entropy of egonet	0.078
% of reaction obtained from egonet (Figure 9.1)	**0.043**
Correlation of user emotion with 1-hop network (Table 9.2)	**0.00713**
Correlation of user emotion with 2-hop network (Table 9.2)	**0.00904**

Finally, in order to evaluate the differences between the two groups of users with respect to the various behavioral attributes detailed above, we perform a test of statistical significance, shown in Table 9.3. We find that the difference between the two groups is statistically significant with respect to most attributes.

9.5 Predictive Model for Depression

Differential Analysis: Based on our investigation thus far, we now conduct a differential analysis to identify features that can be used to distinguish between the behavior of users that potentially suffer from clinical depression and those that do not.

In Figure 9.6, we plot a heat map distinguishing online behavioral features of depressed class users from non-depressed class users. The y-axis shows a selected sample of users belonging to the depressed and normal classes. We explore the following contextual, linguistic and structural and network engagement based features representative to their behavioral identity on social media (displayed on the x-axis): average textual emotion expressed, user clustering coefficient with respect to their one-hop network, linguistic style

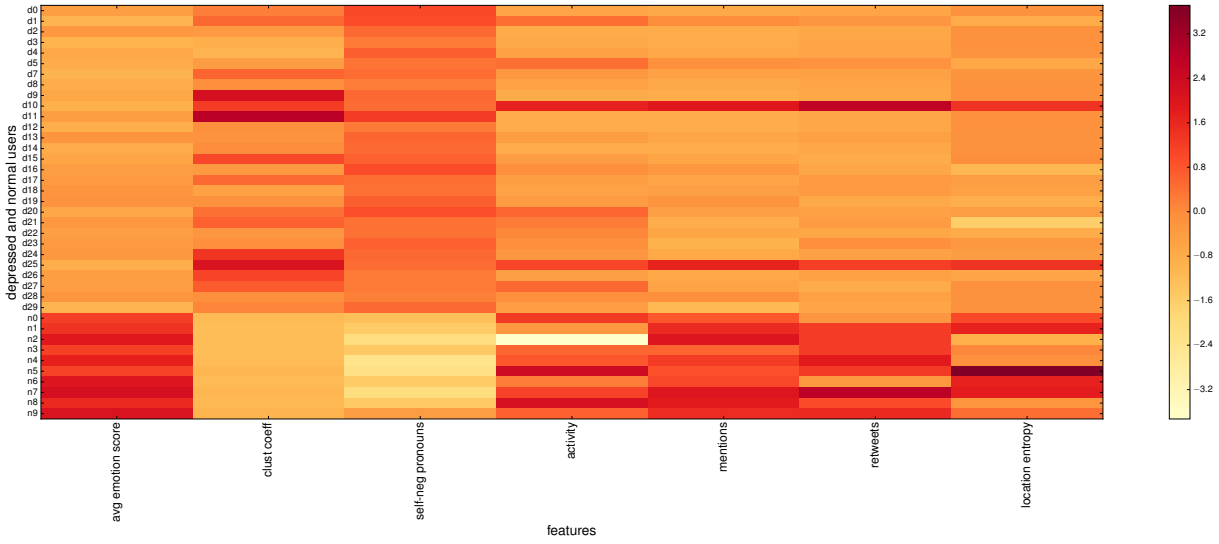


Figure 9.6: Heatmap distinguishing behavioral features of depressed users from non-depressed users. In the interests of space, we present results over a representative of non-depressed and depressed users (the same sample as Figure 9.2).

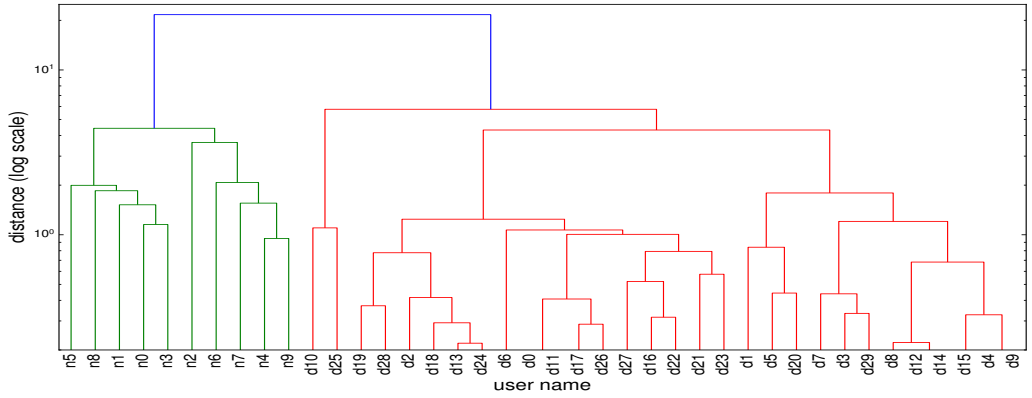


Figure 9.7: Dendrogram of depressed and normal users based on hierarchical clustering of user features. In the interests of space, we present results over a representative of non-depressed and depressed users (the same sample as Figure 9.2).

(the proportion of self and negativity related pronoun usage in their text), average user activity (number of posts), average number of mentions and retweets received by the user and the regional entropy of their ego-network. All feature values have been normalized using their z-scores, denoted by the colorbar.

A differential analysis reveals similar feature trends among the depressed class users, that depart from the trends observed for the normal users. Depressed class users exhibit significantly negative emotion in their tweets. They have a higher clustering coefficient than their non-depressed counterparts in most cases. The proportion of self and negative pronoun usage is notably higher in case of the depressed class users. Depressed class users seem to be less active overall than regular users. Further, the number of retweets and mentions received by most depressed class users is clearly lower than those received by the non-depressed class users, showing that the depressed users don't seem to have much of an effect on their ego network. The ego-networks of depressed class users seem more geographically co-located than those of the normal class users.

There are however some users (e.g., $d10$ and $d25$ in Figure 9.6) of the potentially depressed class who are quite different from the rest of their class members. They seem to be more similar to the normal class of users in features such as activity, mentions, retweets and network location entropy. They are more active online and receive positive reinforcement from their network in the form of an increased number of mentions and retweets.

Building Predictive Models: Since the features identified above discriminate quite well between the two classes of users, we utilize them in order to predict depression in our two user classes. For this, we first inspect how well these features are able to separate the users into 2 classes using Ward's method of agglomerative hierarchical clustering [203]. Figure 9.7 shows this result for a sample of 40 users in the form of a dendrogram, where the

x-axis represents the users and the y axis represents the distance between clusters. We notice that users $d10$ and $d25$ who were identified as somewhat distinct from the rest of their class in Figure 9.6, appear to be more similar to the normal class of individuals here as well.

We subsequently construct a binary classifier using the above identified feature set. We use the Gradient Boosted Decision Trees classifier [86] along with 5-fold cross validation on the full dataset of 150 users. The features distinguishing between the users of the two classes in decreasing order of importance are: words expressing negative emotion, self and group-focused pronouns, user clustering coefficient, activity, retweets, mentions and location entropy. We measure the performance of the classifier using the standard metrics of accuracy of both classes, micro-F1 score and macro F1-score. We achieve an accuracy of 0.9 for the depressed class users, a slightly lower accuracy of 0.87 for the normal/non-depressed class of users, a micro-F1 score of 0.9 and a macro-F1 score of 0.8901. Drilling down on the results, we find that only 5 depressed users are misclassified as normal. Of the misclassified normal users, we find that while their tweets do not express depression, they still exhibit an increasing use of words with negative emotions such as violence or anger.

We additionally use this classification model to predict whether the ego-net of depressed users consists of other similarly depressed users. This would further endorse the theory of clinically depressed users tending to assemble together as a group. While we lack ground truth, we show the users that have been classified as depressed within the one-hop network by the pink colored points in Figure 9.3. The originally depressed and the predicted depressed individuals tend to cluster together with an average clustering coefficient of 0.153, exhibiting some degree of homophily. Many of them are also connected to each other based on their Twitter follower-followee relationship. We validated that these users actually show signs of depression by manually looking at their Twitter feed. We find our results to correlate

quite well with some of the social and psychological research described earlier. In addition, some other network users who did not cluster together with the original depressed class user are also predicted as depressed. As expected, these users are quite far from the core of the network. Some cases (such as user $d5$ in Figure 9.3) do exist who do not cluster together with other similarly depressed users in their network.

9.6 Conclusion

In this chapter, we performed an empirical study on Twitter to understand the online behavior of potentially depressed users against a differential control group of normal users. After building a lexicon of words regularly used in conjunction with clinical depression, we examined a wide range of social media related signals such as linguistic style, emotional signals, user engagement, geo-location and network topology to detect symptomatic cues of depression online. We noticed significant deviations in the behavior of depressed users from the control group in the form of reduced and nocturnal online activity patterns, reduced active and passive network participation, increase in textual negative emotion, distinct linguistic styles (e.g. self-focused pronoun usage), highly clustered and tightly-knit neighborhood topology, a slightly higher geo-location proximity among ego-network members and little to no exchange of influence among depressed users and their ego-network over time. Based on these observations, we extracted relevant features and build a classifier to predict depression among individuals. It achieves an F-score of 90%. Most of our empirical findings corroborated quite well with theoretical literature from the social sciences, medicine and psychology, suggesting that social media interactions may offer a crucial diagnostic tool for clinicians.

Chapter 10: Conclusions, Limitations and Future Work

The rapid and dynamic evolution of technologies and social interactions on the web has given rise to a deluge of human-generated multimodal digital content, across numerous domains, topics and languages. The principal idea behind this dissertation is to develop *efficient and novel algorithms to understand and represent the vast unstructured digital content, as well as the functional or behavioral motives expressed by creators and/or consumers of said content under various contextual settings, in the presence of no or limited human annotated data.* In the previous Chapters 2- 9, we have detailed our main contributions towards the modeling of knowledge and functional intentions for solving the two broad challenges of latent pragmatic analysis, identified in Section 1. In this Chapter, we summarize these contributions, point out their limitations and discuss some promising directions of future research for this dissertation.

10.1 Summarizing our Key Contributions

ETF: Enhancing Taxonomies with Functional Domain Knowledge [339]:

In Chapter 2, we proposed a novel framework, *ETF* (Enhancing Taxonomies with Functional domain knowledge), to automatically enrich large-scale, generic taxonomies with newly emerging domain concepts, avoiding the need for expert or manual curation. For this purpose, we generated a context for each new and existing concept in the taxonomy, from

a corpus of text documents that is associated with them. We trained word-vector embedding models on the existing concepts, and applied these to learn contextual embeddings for the new concepts. We then identified the existing concepts most similar to the new ones, using a nearest neighbor search in the embedding space. We finally learned a ranking model based on background context from the existing taxonomy, and graph-theoretic and semantic features to automatically link the new concepts to similar or related concepts in the current taxonomy. Extensive evaluations of ETF on large, real-world taxonomies of Wikipedia and WordNet showcased significant improvements compared to state-of-the-art baselines.

BOLT-K: Bootstrapping Ontology Learning via Transfer of Knowledge [337]:

In Chapter 3, we proposed a flexible and generalizable framework *BOLT-K* (Bootstrapping Ontology Learning via Transfer of Knowledge), to automatically learn structured ontologies for contemporary newly emergent topics or sub-domains of rapidly evolving fields such as bio-medicine, epidemiology, and e-commerce. For this, we first obtained all the concepts that would constitute the *target* domain ontology. Due to lack of expert-labeled training data for the new target domain, we utilized publicly available textual corpora, and textual data augmentation techniques to generate sufficient labeled training data. We employed semantic and topological attributes to identify the target domain concept pairs likely to be ontologically related, and eliminate the remaining spurious combinations. We finally learnt an LSTM neural network model with attentive pooling, in a multi-task learning fashion. We trained our model jointly on the target labeled training data as well as existing knowledge adapted and *transferred* from a functionally or semantically related *source* domain ontology. Finally, we predicted ontological relationships between the concepts of the target domain, to construct an ontology for it. We extensively evaluated BOLT-K on several real-world

datasets, (i) highlighting the transferability of concepts across comparable sub-domains, and (ii) detecting novel types of relationships that were unseen during training.

FACE-KEG: Fact Checking Explained using Knowledge Graphs [342]:

In Chapter 4, we proposed a framework *FACE-KEG* (FAct Checking Explained using KnowledgE of Graphs), to perform *explainable fact checking* automatically. It consisted of an encoder-decoder setup to detect if a given claim is factually correct or incorrect, by jointly modeling structured and unstructured contextual knowledge associated with the claim. FACE-KEG further learned to generate textual explanations justifying the veracity or falsity of the input claims from the interrelated perspectives of the claim content, suitable background context and structured conceptual knowledge relevant to the claim. As per our knowledge, this was the *first* attempt in the literature to explain fact checking by directly generating human-readable textual explanations for input facts. FACE-KEG first built a knowledge graph and retrieved unstructured background context pertinent to each input claim. A novel graph transformer network and a bidirectional RNN were employed to encode the knowledge graph and textual context respectively. This was followed by jointly training a classifier that predicted if the claim is true or not, and a decoder that learned to generate an abstractive natural language explanation clarifying the veracity or falsity of the claim. We extensively evaluated our approach using a mixture of both automated and human evaluation measures, and achieved significant gains over state-of-the-art baselines.

OPINE: Open Intent Extractions from Natural Language Interactions [185, 336]:

In Chapter 5, we defined a novel problem of *open intent discovery*, and proposed a neural network based framework called *OPINE* (OPen INtent Extraction) to solve it. The

goal was to automatically discover actionable user intents explicitly mentioned in natural language, *without* prior knowledge of a comprehensive list of intent classes that the text utterance may comprise of. In other words, OPINE could recognize instances of intent types that it had never seen before. Unlike prior literature, our proposed approach modeled the open intent discovery problem as a *sequence tagging* task. We developed a neural model consisting of a Conditional Random Field (CRF) on top of a bidirectional LSTM with a multi-head self-attention mechanism. We further employed adversarial training at the lower layers of our model, and unsupervised pre-training in the target domain under consideration. Extensive experiments on multiple real-world datasets showed the accuracy and efficacy of our approach. Moreover, commonly used intent-labeled datasets in dialog research such as SNIPS [61] or ATIS [62, 122] largely have concise, coherent and single-sentence texts. They are not very representative of complex, real-world dialog scenarios which could be verbose and ungrammatical, with intents scattered throughout their content. Therefore, we developed a large, intent-annotated dataset with 25K real-world utterances from the online question-answer forum of Stack Exchange.

ADVIN: Automatically Discovering Novel Domains and Intents from User Utterances [335]:

Our previous work on OPINE could only identify user utterances containing *actionable* intents, and also could not identify the domains of the input text utterances. In Chapter 6, we sought to tackle these two limitations, by solving the problem of novel user intent and domain discovery. We attempted to bridge the gap between the two challenging yet realistic tasks of (i) identifying utterances belonging to novel, generic intents and/or domains, not seen before during model training, and (ii) organizing the newly discovered intents and domains into a taxonomy. We proposed a novel, three-step framework called *ADVIN*

(Automated Discovery of noVel domaIns and iNtents). It automatically discovered user intents and domains in massive, unlabeled text corpora, *without* any prior knowledge about the intents or domains that the text may comprise of. Our method first leveraged the pre-trained multi-layer transformer network, BERT [69], to determine if an utterance is likely to contain a novel intent or not. ADVIN next used unsupervised knowledge transfer to discover the latent intent categories in the earlier identified utterances. Finally, ADVIN hierarchically linked semantically related groups of newly discovered intents to form new domains. We extensively evaluated ADVIN on four public benchmark datasets and real-world data from a commercial voice agent, and significantly outperformed baselines across various empirical configurations.

Multimodal Analysis of Digital Media Content in the Advertising Domain [345, 347]:

In Chapter 7, we proposed and implemented a computational framework that analyzes digital video content in a multimodal fashion, utilizing three modalities of video, audio and natural language. In particular, we performed a predictive analysis of content-based features extracted from online commercial advertisement videos. We extracted multi-dimensional temporal patterns from advertisement videos using multimedia signal processing and natural language processing tools. Features from the three modalities were employed to train separate LSTM neural network models. These models were then fused together to learn a shared representation, for cross modality feature learning. Subsequently, another LSTM model trained on this joint representation was utilized as a classifier to predict effectiveness of the input advertisement among its target audience group. We validated our approach using subjective ratings from a dedicated user study, the text sentiment strength of online viewer comments, and a viewer opinion metric of the likes/views ratio of each advertisement

from the video-sharing website *YouTube*. We investigated the interplay of complex factors contributing to advertisement success among viewers such as the mix of reason and emotion, synergistic interactions between music and narrative speech, spatio-temporal organization of video shots, and brand label mentions. We elicited a useful set of auditory, visual and linguistic patterns to aid advertisement effectiveness, and the design and production of commercial advertisements.

Predicting Trust Relationships in Online Users [343, 344]:

Seeking explicit assessments of trust among users at scale is impractical in an online social network setting. Instead, grounded in appropriate social and psychological theories, in Chapter 8, we developed an unsupervised model to learn a representation for the human socio-behavioral trait of trust in an online social network setting. We integrated the implicit factors of social influence exerted by each user over their social network, users' inherent structural roles obtained from their underlying network topology, and the semantic intentions and affective valence extracted from their short, informal, asynchronous texts. A key finding was the importance of modeling influence and affective valence in such exchanges and their role in detecting stable trust relationships. We extensively evaluated these ideas and demonstrated significant gains over competitive baselines across multiple social media datasets drawn from various scenarios.

Emotional and Linguistic Cues of Online Behavior Patterns Focused on Clinical Depression [341]:

In Chapter 9, we investigated if online communication among users and their social network neighborhood exhibited behavioral patterns similar to those from offline social

engagement, with a focus on *clinical depression*. To facilitate the analysis of our observational study, we examined network effects related to user participation, user engagement and interactions with their ego-neighborhood. We defined network participation features to include both passive (content users are exposed to or receive from other users) and active participation (content created by users themselves). We defined network experience features to include both content (e.g., linguistic cues, emotion) and relational dynamics (e.g., conflict/support, influence) of network embedded interactions. We also examined neighborhood effects and analyzed key statistics of the neighborhood such as size, centrality and affinity to form clusters or communities. Our study included both depressed users and their ego-net(s) as well as non-depressed users (control group) and their ego-net(s). We observed significant deviations in the behavior of depressed users from the control group. Based on our observations, we then described an approach to extract relevant features and build a high accuracy classifier to predict the onset of clinical depression in social media users.

10.2 Limitations and Future Work

The long-term research goal of this dissertation is to work towards the problem of latent pragmatic analysis, by analyzing and transforming massive sources of heterogeneous data in diverse contexts, coupled with interdisciplinary domain and user behavior insights in a mutually reinforcing manner. In this last section, we discuss some limitations of our prior contributions and point out some promising directions for future research.

10.2.1 Enriched Representations of Conceptual Knowledge

Our prior work on learning structured knowledge representations from Chapters 2, 3, 4 has depicted information concepts, entities and relationships among them with static and

deterministic representations. A useful future direction is to study the temporal dynamics and consistencies of online information, their impact on existing facts and their correlations with real-world events. For example, the fact *"Barack Obama is the president of the United States" was valid in 2017 but not in 2019).* This includes capturing temporal dependencies among concepts, relationships or facts as a whole. For instance, the fact *"Donald Trump is the president of the United States"* cannot occur temporally 'before' the fact *"Barack Obama is the president of the United States"*. A preliminary example is our work on inspecting the changing online user trust relationships during different temporal phases of a hazard in Chapter 8. For this purpose, we should systematically quantify and incorporate sophisticated linguistic and probabilistic elements into knowledge graph construction and expansion. Some examples include words or phrases indicating uncertainty (e.g. 'maybe', 'barely', 'probably'), sentiments (e.g. 'efficiently', 'accurately'), or qualifiers (e.g. 'much', 'extremely'). Further, our proposed analytical techniques of ETF and BOLT-K that we developed in our past research can only address binary relationship between concepts, i.e. a pair of concept connected by a single relation type. It would be interesting to extend our past work to model *n*-ary concept relationships (e.g. protein localization relations), while preserving mutual constraints between relations and their corresponding concepts.

As mentioned earlier, people or humans represent an important facet of online information, as its authors, audience and often subjects of its content. Therefore, an interesting thread of future investigation lies in augmenting existing structured knowledge repositories with associated socio-cultural user behavior information, to yield a more nuanced and holistic understanding of the world. The developed algorithms must address the challenges that will inevitably arise from modeling, learning, and inference using such rich, complex human social interactions. For instance, structured knowledge graphs constructed based

on mined user behavior facts can be used to predict users' possible subsequent activities or interests based on their current contexts in multiple application scenarios. They can also be used to infer how individuals characterize other individuals or social groups in different ways.

10.2.2 Knowledge Representation Learning for Social Good

Mining knowledge and developing context-aware computational models by levering cues and evidence from other disciplines such as psychology, social science and bio-medicine has been a key focus of this dissertation. To this end, an interesting future direction is to study ways of integrating heterogeneous unstructured (e.g. surveys, inventories, maps, social media data) and structured (e.g. knowledge bases, ontologies) sources of information in the humanitarian domain. We made initial attempts in this direction in our explainable fact checking technique FACE-KEG (Chapter 4), by jointly utilizing unstructured and structured information. This is a challenging problem due to the issues of data representation, aggregating individual level behaviors, and unambiguous alignment of concepts across several data sources.

Our earlier work on understanding user behavioral characteristics and user intent (Chapters 5, 6, 8, 9) further has the scope to be extended. The behavioral, linguistic and network based signals identified from our work on clinical depression in Chapter 9 can provide adaptive and supportive content updates to online users, especially those afflicted by mental disorders or social discrimination. This can include sharing helpful content from *trustworthy* peers in their ego-network (based on our work in Chapter 8), or automatically learning to generate supportive and empathetic text to encourage them.

Another natural extension of our work studying online human behavior characteristics that we did not explore is investigating more complex, community-level aspects of social good, e.g. *collective efficacy*[16]. Such studies can reveal nuanced insights into the intentions and interactions of online community users. They can also facilitate downstream tasks like recommending friends and content, supporting minorities, detecting temporal evolution of group behavior etc. Such research can usefully benefit from other disciplines, such as psycho-linguistics, psychology, and cognitive science, apart from computer science.

10.2.3 Human-in-the-loop Learning

Receiving different forms of human feedback or guidance either from experts or crowd sourcing platforms can serve as powerful sources of supervision and contextual knowledge at various stages of algorithm development. Much of our past research has only incorporated human feedback at the evaluation stage. Considering this, a logical future direction of research is to examine machine learning approaches that intelligently interact with and dynamically learn from humans in a more direct manner during model development (*human-in-the-loop* learning). We did this at a preliminary level in our work on automatically discovering novel domains and intents from text utterances (ADVIN in Chapter 6). We solicited human feedback regarding the granularity (e.g. too specific or too generic) of the grouping of newly discovered intents to form domains. Our ADVIN model was subsequently updated in a semi-supervised manner to generate revised domains that better matched the users' mental model and current information/application needs, and increased the user confidence in the results of our model. A principled analysis of effective formats of collecting human guidance depends on the problem context as well as the background and

[16]Social cohesion among neighbors combined with their willingness to intervene on behalf of the common good [274]

demographics of the humans involved, and is an interesting problem worth pursuing. A possible way of seeking human intervention during model training could be via a model-based intelligent agent. It would iteratively employ a learning algorithm to decide whether human feedback is needed at that point, and automatically request it in a user-friendly manner. The obtained feedback can then be used to update the agent's current state.

10.2.4 Multimodal Investigation of User Intent in Heterogeneous Environments

Social media has increasingly become a rich resource carrying user intent information ranging from technical support conversations with commercial companies, to requests for assistance during an emergency or crisis. It can be quite valuable to be able to be recognize user intentions from their social media posts. Our proposed prior approaches for intent discovery (Chapters 5 and 6) are tailored to relatively clean text with minimal grammatical or spelling errors, and largely utilize only the textual context of the input text. We do not consider the plethora of data from other modalities that people also utilize to convey their intentions, such as audio or speech characteristics derived from user interactions with virtual assistants (e.g. Amazon Alexa, Microsoft Cortana), human gestures from device interactions, images posted online etc. It would be worth extending our work of multimodal content analysis (similar to what we proposed in Chapter 7 in the video advertising domain), to the human intent detection problem space. Further, there is a rich repertoire of features apart from the content semantics that we did not investigate, which can help to develop a more nuanced model for intent detection. Some examples include user demographics, click behavior, users' historical behavior, geo-location, informal language patterns, emojis, user engagement, sentiment or affect and language stylistics.

There are a couple of additional connected lines of research in this domain that we did not look into in this dissertation, but are worth pursuing. First, there is a notion of a *personalized intent* for each user. For example, two users sharing the same social media post may have totally different intents they would like to express, based on their opinions, cognitive abilities or personality traits. An related idea is to abstract and transfer our knowledge of the latent intents of a user in a particular domain to infer their preferences in other, *similar* domains. Second, most of the deep neural network models that we previously developed are expensive to maintain in terms of computational and memory resources. This might hinder their deployment in devices with low memory or strict latency requirements (e.g. personal assistants such as Apple Siri, Amazon Alexa). Thus, it could be meaningful to study how to perform *model compression* and *acceleration* in deep networks without significantly affecting model accuracy or performance.

Bibliography

[1] Sibel Adali, Robert Escriva, Mark K Goldberg, Mykola Hayvanovych, Malik Magdon-Ismail, Boleslaw K Szymanski, William A Wallace, and Gregory Williams. Measuring behavioral trust in social networks. In *IEEE ISI*, 2010.

[2] Sibel Adali, Fred Sisenda, and Malik Magdon-Ismail. Actions speak as loud as words: Predicting relationships from social behavior data. In *WWW*, 2012.

[3] Swati Agarwal and Ashish Sureka. Characterizing linguistic attributes for automatic classification of intent based racist/radicalized posts on tumblr micro-blogging website. *arXiv preprint arXiv:1701.04931*, 2017.

[4] Ferit Akova, Murat Dundar, Yuan Qi, and Bartek Rajwa. Self-adjusting models for semi-supervised learning in partially observed settings. In *IEEE ICDM*, 2012.

[5] U. Alon, S. Brody, O. Levy, and E. Yahav. code2seq: Generating sequences from structured representations of code. *arXiv preprint 1808.01400*, 2018.

[6] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853, 2005.

[7] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 344–354, 2015.

[8] Cigdem Aslay, Nicola Barbieri, Francesco Bonchi, and Ricardo A Baeza-Yates. Online topic-aware influence maximization queries. In *EDBT*, pages 295–306, 2014.

[9] Sitaram Asur and Srinivasan Parthasarathy. A viewpoint-based approach for interaction graph analysis. In *SIGKDD*, 2009.

[10] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. Generating fact checking explanations. *arXiv preprint arXiv:2004.05773*, 2020.

[11] Ching-man Au Yeung and Tomoharu Iwata. Strength of social influence in trust networks in product review sites. In *WSDM*, 2011.

[12] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*. 2007.

[13] I. Augenstein, C. Lioma, D. Wang, L. C. Lima, C. Hansen, C. Hansen, and J. G. Simonsen. Multifc: A real-world multi-domain dataset for evidence-based fact checking of claims. *arXiv preprint arXiv:1909.03242*, 2019.

[14] R. Baly, M. Mohtarami, J. Glass, L. Màrquez, A. Moschitti, and P. Nakov. Integrating stance detection and fact checking in a unified corpus. *arXiv preprint 1804.08012*, 2018.

[15] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.

[16] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Topic-aware social influence propagation models. *Knowledge and information systems*, 2013.

[17] D. Beck, G. Haffari, and T. Cohn. Graph-to-sequence learning using gated graph neural networks. *arXiv preprint 1806.09835*, 2018.

[18] Ghazaleh Beigi, Jiliang Tang, Suhang Wang, and Huan Liu. Exploiting emotional information for trust/distrust prediction. In *SIAM*, 2016.

[19] Luisa Bentivogli, Andrea Bocco, and Emanuele Pianta. Archiwordnet: integrating wordnet with domain-specific knowledge. In *Global Wordnet Conference*, 2004.

[20] Joyce Berg, John Dickhaut, and Kevin McCabe. Trust, reciprocity, and social history. *Games and economic behavior*, 1995.

[21] Aditya Bhargava, Asli Celikyilmaz, Dilek Hakkani Tür, and Ruhi Sarikaya. Easy contextual intent prediction and slot detection. In *ICASSP*, 2013.

[22] David M Blei. Probabilistic topic models. *Communications of the ACM*, 2012.

[23] Bruce Block. *The Visual Story: Creating the Visual Structure of Film, TV and Digital Media*. CRC Press, 2013.

[24] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD*, 2008.

[25] Johan Bollen, Huina Mao, and Alberto Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *ICWSM*, 2011.

[26] Claire Bonial, Julia Bonn, Kathryn Conger, Jena D Hwang, and Martha Palmer. Propbank: Semantics of new predicate types. In *LREC*, pages 3013–3019, 2014.

[27] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013.

[28] Margaret M Bradley and Peter J Lang. Affective norms for english words. Technical report, C-1, the center for research in psycho physiology, University of Florida, 1999.

[29] Ulrik Brandes and Daniel Fleischer. Centrality measures based on current flow. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 533–544. Springer, 2005.

[30] Gordon C Bruner. Music, mood, and marketing. *Journal of marketing*, 1990.

[31] Wilma Bucci and Norbert Freedman. The language of depression. *Bulletin of the Menninger Clinic*, 1981.

[32] M. Buhrmester, T. Kwang, and S. Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality data? 2016.

[33] Paul Buitelaar and Bogdan Sacaleanu. Extending synsets with medical terms. In *Proceedings of the International WordNet Conference*, 2002.

[34] C.J.C. Burges. From ranknet to lambdarank to lambdamart: An overview. Technical report, 2010.

[35] John T Cacioppo and Richard E Petty. The elaboration likelihood model of persuasion. *NA-Advances in Consumer Research Volume 11*, 1984.

[36] D. Cai and W. Lam. Graph transformer for graph-to-sequence learning. In *AAAI*, 2020.

[37] Ruichu Cai, Binjun Zhu, Lei Ji, Tianyong Hao, Jun Yan, and Wenyin Liu. A cnn-lstm attention approach to understanding user query intent from online health communities. In *IEEE ICDMW Workshops*, 2017.

[38] Joseph Campbell. *The Hero With a Thousand Faces*. New World Library, 2008.

[39] Margaret C Campbell and Kevin Lane Keller. Brand familiarity and advertising repetition effects. *Journal of Consumer Research*, 2003.

[40] Kathi Canese and Sarah Weis. Pubmed: the bibliographic database. 2013.

[41] Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. Multi-lingual intent detection and slot filling in a joint bert-based model. *arXiv preprint arXiv:1907.02884*, 2019.

[42] Carlos Castillo. *Big Crisis Data*. Cambridge University Press, 2016.

[43] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and P Krishna Gummadi. Measuring user influence in twitter: The million follower fallacy. *ICWSM*, 2010.

[44] T. Chakrabarty, T. Alhindi, and S. Muresan. Robust document retrieval and individual evidence modeling for fact extraction and verification. In *Workshop on Fact Extraction and VERification (FEVER)*, 2018.

[45] Stevie Chancellor, Zhiyuan Lin, Erica L Goodman, Stephanie Zerwas, and Munmun De Choudhury. Quantifying and predicting mental illness severity in online pro-eating disorder communities. In *ACM CSCW*, 2016.

[46] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, 2002.

[47] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[48] D. Chen, A. Fisch, J. Weston, and A. Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint 1704.00051*, 2017.

[49] H. Chen, S. Huang, D. Chiang, and J. Chen. Improved neural machine translation with a syntax-aware encoder and decoder. *arXiv preprint 1707.05436*, 2017.

[50] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen. Enhanced lstm for natural language inference. *arXiv preprint 1609.06038*, 2016.

[51] W. Chen, H. Wang, J. Chen, Y. Zhang, H. Wang, S. Li, X. Zhou, and W. Wang. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint 1909.02164*, 2019.

[52] Wei Chen, Tian Lin, and Cheng Yang. Efficient topic-aware influence maximization using preprocessing. *CoRR, abs/1403.0057*, 2014.

[53] Yu Chen and Mohammed J Zaki. Kate: K-competitive autoencoder for text. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 85–94. ACM, 2017.

[54] Zhiyuan Chen, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Identifying intention posts in discussion forums. In *NAACL*, 2013.

[55] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*, 2015.

[56] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 1960.

[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*, 12(Aug):2493–2537, 2011.

[58] Ryan Compton, David Jurgens, and David Allen. Geotagging one hundred million twitter accounts with total variation minimization. In *IEEE Big Data*, 2014.

[59] Glen Coppersmith, Mark Dredze, and Craig Harman. Quantifying mental health signals in twitter. 2014.

[60] Glen Coppersmith, Mark Dredze, Craig Harman, and Kristy Hollingshead. From adhd to sad: Analyzing the language of mental health on twitter through self-reported diagnoses. *NAACL HLT*, 2015.

[61] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, et al. Snips voice platform. *arXiv preprint arXiv:1805.10190*, 2018.

[62] Deborah A Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, et al. Expanding the scope of the atis task: The atis-3 corpus. In *HLT Workshop*, 1994.

[63] Samuel B Day and Robert L Goldstone. The import of knowledge export: Connecting findings and theories of transfer of learning. *Educational Psychologist*, 47(3):153–176, 2012.

[64] Munmun De Choudhury, Scott Counts, and Eric Horvitz. Predicting postpartum changes in emotion and behavior via social media. In *ACM SIGCHI*, 2013.

[65] Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. Predicting depression via social media. In *ICWSM*, 2013.

[66] Munmun De Choudhury, Emre Kiciman, Mark Dredze, Glen Coppersmith, and Mrinal Kumar. Discovering shifts to suicidal ideation from mental health content in social media. In *ACM SIGCHI*, 2016.

[67] John Deighton, Daniel Romer, and Josh McQueen. Using drama to persuade. *Journal of Consumer research*, 1989.

[68] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, 2017.

[69] J. Devlin, M. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint 1810.04805*, 2018.

[70] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *SIGKDD*, 2001.

[71] Xin Luna Dong. Challenges and innovations in building a product knowledge graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2869–2869. ACM, 2018.

[72] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 766–774, 2014.

[73] Murat Dundar, Ferit Akova, Alan Qi, and Bartek Rajwa. Bayesian nonexhaustive learning for online discovery and modeling of emerging classes. *arXiv preprint arXiv:1206.4600*, 2012.

[74] Larry Elin and Alan Lapides. *Designing and Producing the Television Commercial*. Pearson, 2004.

[75] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. *arXiv preprint arXiv:1705.00440*, 2017.

[76] Rino Falcone and Cristiano Castelfranchi. Social trust: A cognitive approach. In *Trust and deception in virtual societies*. 2001.

[77] Michael Färber, Achim Rettinger, and Boulos El Asmar. On emerging entity detection. In *European Knowledge Acquisition Workshop*, pages 223–238. Springer, 2016.

[78] Christiane Fellbaum, Udo Hahn, and Barry Smith. Towards new information resources for public health. *Journal of Biomedical Informatics*, 2006.

[79] WordNet Fellbaum. An electronic lexical database (language, speech, and communication), 1998.

[80] V. Fionda and G. Pirrò. Fact checking via evidence patterns. In *IJCAI*, 2018.

[81] Mauajama Firdaus, Shobhit Bhatnagar, Asif Ekbal, and Pushpak Bhattacharyya. A deep learning based multi-task ensemble model for intent detection and slot filling in spoken language understanding. In *International Conference on Neural Information Processing*, 2018.

[82] BJ Fogg and Hsiang Tseng. The elements of computer credibility. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 80–87. ACM, 1999.

[83] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3), 1973.

[84] Santo Fortunato. Community detection in graphs. *Physics reports*, 2010.

[85] James H Fowler and Nicholas A Christakis. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study. *British Medical Journal*, 2008.

[86] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 2002.

[87] Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. Learning semantic hierarchies via word embeddings. In *ACL Vol 1 Long Papers*, 2014.

[88] Yanwei Fu and Leonid Sigal. Semi-supervised vocabulary-informed learning. In *CVPR*, 2016.

[89] M. Gad-Elrab, D. Stepanova, J. Urbani, and G. Weikum. Exfakt: A framework for explaining facts over knowledge graphs and text. In *ACM WSDM*, 2019.

[90] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[91] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[92] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. In *Workshop for NLP Open Source Software (NLP-OSS)*, 2018.

[93] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, 1999.

[94] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of AISTATS*, 2011.

[95] Jennifer Golbeck. Trust and nuanced profile similarity in online social networks. *TWEB*, 2009.

[96] Jennifer Golbeck and James Hendler. Filmtrust: Movie recommendations using trust in web-based social networks. In *IEEE CCNC*, 2006.

[97] Jennifer Golbeck and James Hendler. Inferring binary trust relationships in web-based social networks. *TOIT*, 2006.

[98] Debra S Goldberg and Frederick P Roth. Assessing experimentally derived interactions in a small world. *Proceedings of the National Academy of Sciences*, 2003.

[99] Scott A Golder and Michael W Macy. Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science*, 2011.

[100] Jingjing Gong, Xipeng Qiu, Shaojing Wang, and Xuanjing Huang. Information aggregation via dynamic routing for sequence encoding. *arXiv preprint arXiv:1806.01501*, 2018.

[101] Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. Slot-gated modeling for joint slot filling and intent prediction. In *NAACL*, 2018.

[102] Ian J Goodfellow, Jonathon Shlens, and Christian E Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[103] K Gowda and G Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition*, 1978.

[104] Nicolai Graakjaer. *Analyzing Music in Advertising: Television Commercials and Consumer Choice*. Routledge, 2014.

[105] Mark S Granovetter. The strength of weak ties. *American journal of sociology*, 1973.

[106] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *IEEE ICASSP*, 2013.

[107] Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *WWW*, 2004.

[108] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith. Etaf: An extended trust antecedents framework for trust prediction. In *ASONAM*, 2014.

[109] Z. Guo, Y. Zhang, Z. Teng, and W. Lu. Densely connected graph convolutional networks for graph-to-sequence learning. *TACL*, 2019.

[110] Abhishek Gupta, Yew-Soon Ong, and Liang Feng. Insights on transfer optimization: Because experience is the best teacher. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):51–64, 2017.

[111] Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. Semantic parsing for task oriented dialog using hierarchical representations. *arXiv preprint arXiv:1810.07942*, 2018.

[112] Vineet Gupta, Devesh Varshney, Harsh Jhamtani, Deepam Kedia, and Shweta Karwa. Identifying purchase intent from social posts. In *ICWSM*, 2014.

[113] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[114] A. Hanselowski, H. Zhang, Z. Li, D. Sorokin, B. Schiller, C. Schulz, and I. Gurevych. Ukp-athene: Multi-sentence textual entailment for claim verification. *arXiv preprint 1809.01479*, 2018.

[115] S.M. Harabagiu, S.J. Maiorano, and M.A. Paşca. Open-domain textual question answering techniques. *Natural Language Engineering*, 2003.

[116] Silvana Hartmann, Ilia Kuznetsov, Teresa Martin, and Iryna Gurevych. Out-of-domain FrameNet semantic role labeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 471–482, Valencia, Spain, April 2017. Association for Computational Linguistics.

[117] Carleen Hawn. Take two aspirin and tweet me in the morning: how twitter, facebook, and other social media are reshaping health care. *Health Affairs*, 2009.

[118] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE ICCV*, 2015.

[119] Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. Deep semantic role labeling: What works and what's next. In *ACL*, 2017.

[120] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 507–517. International World Wide Web Conferences Steering Committee, 2016.

[121] Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *ACL COLING*, 1992.

[122] Charles Hemphill, John Godfrey, and George Doddington. The atis spoken language systems pilot corpus. In *a Workshop on Speech and Natural Language*, 1990.

[123] S. Hochreiter and J. Schmidhuber. Long short-term memory. In *Neural computation*, 1997.

[124] Aaron Hogue and Laurence Steinberg. Homophily of internalized distress in adolescent peer groups. *Developmental Psychology*, 1995.

[125] A. Hotho, A. Maedche, and S. Staab. Ontology-based text document clustering. *KI*, 2002.

[126] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[127] Cho-Jui Hsieh, Kai-Yang Chiang, and Inderjit S Dhillon. Low rank modeling of signed networks. In *ACM SIGKDD*, 2012.

[128] Yen-Chang Hsu and Zsolt Kira. Neural network-based clustering using pairwise constraints. *arXiv preprint arXiv:1511.06321*, 2015.

[129] http://wiki.dbpedia.org/dbpedia-version 2015-10. Dbpedia version, 2015.

[130] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. *arXiv preprint arXiv:1703.00955*, 2017.

[131] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[132] I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene. Unsupervised graph-based topic labelling using dbpedia. In *ACM WSDM*, 2013.

[133] Rick E Ingram, Debra Cruet, Brenda R Johnson, and Kathleen S Wisnicki. Self-focused attention, gender, gender role, and vulnerability to negative affect. *Journal of Personality and Social Psychology*, 1988.

[134] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*, 2018.

[135] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *TOIS*, 2002.

[136] Lauren A Jelenchick, Jens C Eickhoff, and Megan A Moreno. "facebook depression?" social networking site use and depression in older adolescents. *Journal of Adolescent Health*, 2013.

[137] Minwoo Jeong and Gary Geunbae Lee. Triangular-chain conditional random fields. *IEEE TASLP*, 2008.

[138] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 687–696, 2015.

[139] Thorsten Joachims. Training linear svms in linear time. In *ACM SIGKDD*, 2006.

[140] Charles Jochim and Lea Deleris. Named entity recognition in the medical domain with constrained crf models. In *EACL*, 2017.

[141] Thomas E Joiner, Mark S Alfano, and Gerald I Metalsky. When depression breeds contempt: Reassurance seeking, self-esteem, and rejection of depressed college students by their roommates. *Journal of Abnormal Psychology*, 1992.

[142] Daniel Jurafsky. Pragmatics and computational linguistics. *Handbook of pragmatics*, 2004.

[143] D. Jurgens and M.T. Pilehvar. Reserating the awesometastic: An automatic extension of the wordnet taxonomy for novel terms. In *NAACL*, 2015.

[144] David Jurgens and Mohammad Taher Pilehvar. Semeval-2016 task 14: Semantic taxonomy enrichment. In *SemEval@ NAACL-HLT*, 2016.

[145] David Jurgens and Mohammad Taher Pilehvar. Semeval-2016 task 14: semantic taxonomy enrichment. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1092–1102, 2016.

[146] Kushal Kafle, Mohammed Yousefhussien, and Christopher Kanan. Data augmentation for visual question answering. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 198–202, 2017.

[147] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW*, 2003.

[148] Leo Katz. A new status index derived from sociometric analysis. In *Psychometrika*, 1953.

[149] Yarden Katz and Jennifer Golbeck. Social network-based trust in prioritized default logic. In *AAAI*, 2006.

[150] Ichiro Kawachi and Lisa F Berkman. Social ties and mental health. *Journal of Urban Health*, 2001.

[151] Kevin Lane Keller et al. *Strategic Brand Management*. Upper Saddle River, NJ: Prentice Hall, 1998.

[152] Kari Kelton, Kenneth R Fleischmann, and William A Wallace. Trust in digital information. *JASIST*, 2008.

[153] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, 2003.

[154] J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. Genia corpus: a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182, 2003.

[155] Joo-Kyung Kim and Young-Bum Kim. Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisficing false acceptance rates. *arXiv preprint arXiv:1807.00072*, 2018.

[156] Joo-Kyung Kim, Gokhan Tur, Asli Celikyilmaz, Bin Cao, and Ye-Yi Wang. Intent detection using semantically enriched word embeddings. In *IEEE SLT*, 2016.

[157] Young-Bum Kim, Dongchan Kim, Anjishnu Kumar, and Ruhi Sarikaya. Efficient large-scale neural domain classification with personalized attention. In *Proceedings of ACL*, 2018.

[158] Young-Bum Kim, Sungjin Lee, and Karl Stratos. Onenet: Joint domain, intent, slot prediction for spoken language understanding. In *ASRU*, 2017.

[159] Young-Bum Kim, Karl Stratos, and Dongchan Kim. Adversarial adaptation of synthetic or stale data. In *Proceedings of ACL*, 2017.

[160] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint 1412.6980*, 2014.

[161] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 1999.

[162] R. Koncel-Kedziorski, D. Bekal, Y. Luan, M. Lapata, and H. Hajishirzi. Text generation from knowledge graphs with graph transformers. *arXiv preprint 1904.02342*, 2019.

[163] I. Konstas, S. Iyer, M. Yatskar, Y. Choi, and L. Zettlemoyer. Neural amr: Sequence-to-sequence models for parsing and generation. *arXiv preprint 1704.08381*, 2017.

[164] Raghavendra Kotikalapudi, S Chellappan, F Montgomery, D Wunsch, and K Lutzen. Associating depressive symptoms in college students with internet usage using real internet data. *IEEE Technology and Society Magazine*, 2012.

[165] Erwin Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, 1979.

[166] Trausti Kristjansson, Aron Culotta, Paul Viola, and Andrew McCallum. Interactive information extraction with constrained conditional random fields. In *AAAI*, 2004.

[167] Anjishnu Kumar, Pavankumar Reddy Muddireddy, Markus Dreyer, and Björn Hoffmeister. Zero-shot learning across heterogeneous overlapping domains. In *INTERSPEECH*, 2017.

[168] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *WWW*, 2010.

[169] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[170] P. Lahoti, P.K. Nicholson, and B. Taneva. Efficient set intersection counting algorithm for text similarity measures. In *ALENEX*, 2017.

[171] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, 2015.

[172] Christian A Larsen. *The rise and fall of social cohesion: The construction and deconstruction of social trust in the US, UK, Sweden and Denmark*. Oxford University Press, 2013.

[173] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, 2014.

[174] Ig-hoon Lee, Suekyung Lee, Taehee Lee, Sang-goo Lee, Dongkyu Kim, Jonghoon Chun, Hyunja Lee, and Junho Shim. Practical issues for building a product ontology system. In *Proceedings of the International Workshop on Data Engineering Issues in E-Commerce*, pages 16–25. IEEE, 2005.

[175] Geoffrey N Leech. *Principles of pragmatics*. Routledge, 2016.

[176] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, 2010.

[177] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*, 2017.

[178] Michael S Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM TOMM*, 2006.

[179] Baichuan Li, Jing Liu, Chin-Yew Lin, Irwin King, and Michael R Lyu. A hierarchical entity-based approach to structuralize user generated content in social media: A case of yahoo! answers. In *EMNLP*, 2013.

[180] C. Lin and E. Hovy. Manual and automatic evaluation of summaries. In *ACL Workshop on Automatic Summarization*, 2002.

[181] Ting-En Lin and Hua Xu. Deep unknown intent detection with margin loss. *arXiv preprint arXiv:1906.00434*, 2019.

[182] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 2015.

[183] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2124–2133, 2016.

[184] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[185] Nedim Lipka and Nikhita Vedula. Utilizing recurrent neural networks to recognize and extract open intent from text inputs. *US Patent App. 16/216,296*, 2020.

[186] Bing Liu and Ian Lane. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*, 2016.

[187] Bing Liu and Ian Lane. Multi-domain adversarial learning for slot filling in spoken language understanding. *arXiv preprint arXiv:1711.11310*, 2017.

[188] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[189] Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. Benchmarking natural language understanding services for building conversational agents. *arXiv preprint arXiv:1903.05566*, 2019.

[190] Z. Liu, C. Xiong, M. Sun, and Z. Liu. Fine-grained fact verification with kernel graph attention network. In *ACL*, 2020.

[191] Leonard M Lodish, Magid Abraham, Stuart Kalmenson, Jeanne Livelsberger, Beth Lubetkin, Bruce Richardson, and Mary Ellen Stevens. How tv advertising works: A meta-analysis of 389 real world split cable tv advertising experiments. *Journal of Marketing Research*, 1995.

[192] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[193] Albert J Lott and Bernice E Lott. Group cohesiveness as interpersonal attraction: a review of relationships with antecedent and consequent variables. *Psychological bulletin*, 1965.

[194] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*, 2015.

[195] Xinghua Lu, Bin Zheng, Atulya Velivelli, and ChengXiang Zhai. Enhancing text categorization with semantic-enriched representation and training data augmentation. *Journal of the American Medical Informatics Association*, 13(5):526–535, 2006.

[196] M. Luong, H. Pham, and C. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint 1508.04025*, 2015.

[197] Lisa Lustberg and Charles F Reynolds. Depression and insomnia: questions of cause and effect. *Sleep Medicine Reviews*, 2000.

[198] Fenglong Ma, Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 745–754. ACM, 2015.

[199] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.

[200] L. Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 2008.

[201] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[202] Tiep Mai, Bichen Shi, Patrick K. Nicholson, Deepak Ajwani, and Alessandra Sala. Scalable disambiguation system capturing individualities of mentions. In *LDK*. Springer, 2017.

[203] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer, 2005.

[204] C. Malon. Team papelo: Transformer networks at fever. *arXiv preprint arXiv:1901.02534*, 2019.

[205] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of ACL*, 2014.

[206] D. Marcheggiani and L. Perez. Deep graph convolutional encoders for structured data to text generation. *arXiv preprint 1810.09995*, 2018.

[207] Winter Mason. Crisis mapping as collective problem solving. *ACM Transactions on Applied Perception*, 2010.

[208] Paolo Massa and Paolo Avesani. Controversial users demand local trust metrics: An experimental study on epinions.com community. In *National Conference on AI*, 2005.

[209] Paolo Massa, Kasper Souren, Martino Salvetti, and Danilo Tomasoni. Trustlet, open research on trust metrics. *Scalable Computing: Practice and Experience*, 2001.

[210] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

[211] Suyu Mei, Wang Fei, and Shuigeng Zhou. Gene ontology based transfer learning for protein subcellular localization. *BMC bioinformatics*, 12(1):44, 2011.

[212] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE TASLP*, 2015.

[213] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. Distributed representation of words and phrases and their compositionality. In *NIPS*, 2013.

[214] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL*, 2009.

[215] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.

[216] Takeru Miyato, Andrew Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

[217] SPFGH Moen and Tapio Salakoski2 Sophia Ananiadou. Distributional semantics resources for biomedical text processing. In *Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan*, pages 39–43, 2013.

[218] Brian CJ Moore. *An Introduction to the Psychology of Hearing*. Brill, 2012.

[219] Megan A Moreno, Lauren A Jelenchick, Katie G Egan, Elizabeth Cox, Henry Young, Kerry E Gannon, and Tara Becker. Feeling bad on facebook: depression disclosures by college students on a social networking site. *Depression and Anxiety*, 2011.

[220] Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, et al. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*, 2016.

[221] Meinard Müller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer, 2015.

[222] Mark A Musen, Natalya F Noy, Nigam H Shah, Patricia L Whetzel, Christopher G Chute, Margaret-Anne Story, Barry Smith, and NCBO team. The national center for biomedical ontology. *Journal of the American Medical Informatics Association*, 19(2):190–195, 2011.

[223] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 2007.

[224] Meenakshi Nagarajan, Karthik Gomadam, Amit P Sheth, Ajith Ranabahu, Raghava Mutharaju, and Ashutosh Jadhav. *Spatio-temporal-thematic analysis of citizen sensor data: Challenges and experiences*. 2009.

[225] R. Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 2009.

[226] Roberto Navigli, Paola Velardi, and Stefano Faralli. A graph-based algorithm for inducing lexical taxonomies from scratch. In *IJCAI: International Joint Conference on Artificial Intelligence*, 2011.

[227] Yair Neuman, Yohai Cohen, Dan Assaf, and Gabbi Kedma. Proactive screening for depression through metaphorical and automatic text analysis. *Artificial Intelligence in Medicine*, 2012.

[228] Mark EJ Newman. A measure of betweenness centrality based on random walks. *Social networks*, 2005.

[229] Maximilian Nickel, Lorenzo Rosasco, Tomaso A Poggio, et al. Holographic embeddings of knowledge graphs. In *AAAI Conference on Artificial Intelligence*, volume 2, pages 3–2, 2016.

[230] Y. Nie, H. Chen, and M. Bansal. Combining fact extraction and verification with neural semantic matching networks. In *AAAI*, 2019.

[231] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[232] Jean Nunn, Andrew Mathews, and Peter Trower. Selective processing of concern-related information in depression. *British Journal of Clinical Psychology*, 1997.

[233] Gwenn Schurgin O'Keeffe, Kathleen Clarke-Pearson, et al. The impact of social media on children, adolescents, and families. *Pediatrics*, 127(4):800–804, 2011.

[234] Thomas E Oxman, Stanley D Rosenberg, and Gary J Tucker. The language of paranoia. *American Journal of Psychiatry*, 1982.

[235] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[236] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, pages 1410–1418, 2009.

[237] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[238] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint 1606.01933*, 2016.

[239] Minsu Park, Chiyoung Cha, and Meeyoung Cha. Depressive moods of users portrayed in twitter. In *ACM SIGKDD Workshop on Healthcare Informatics*, 2012.

[240] Michael J Paul and Mark Dredze. You are what you tweet: Analyzing twitter for public health. *ICWSM*, 2011.

[241] Dan Pelleg and Andrew W Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 727–734. Morgan Kaufmann Publishers Inc., 2000.

[242] Nanyun Peng and Mark Dredze. Improving named entity recognition for chinese social media with word segmentation representation learning. *arXiv preprint arXiv:1603.00786*, 2016.

[243] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *arXiv preprint arXiv:1708.03743*, 2017.

[244] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 2001.

[245] James W Pennebaker, Matthias R Mehl, and Kate G Niederhoffer. Psychological aspects of natural language use. *Annual Review of Psychology*, 2003.

[246] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[247] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea. Automatic detection of fake news. *arXiv preprint 1708.07104*, 2017.

[248] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *ACM SIGKDD*, 2014.

[249] Andrew Perrin. Social media usage: 2005-2015. 2015.

[250] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv preprint 1802.05365*, 2018.

[251] Petar Petrovski, Anna Primpeli, Robert Meusel, and Christian Bizer. The wdc gold standards for product feature extraction and product matching. In *International Conference on Electronic Commerce and Web Technologies*, pages 73–86. Springer, 2016.

[252] J. Phang, T. Févry, and S. Bowman. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint 1811.01088*, 2018.

[253] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *World Wide Web Companion*, 2017.

[254] M. Poprat, E. Beisswanger, and U. Hahn. Building a biowordnet by using wordnet's data formats and wordnet's software infrastructure. In *Software engineering, testing, and quality assurance for natural language processing*, 2008.

[255] Hemant Purohit, Yiye Ruan, David Fuhry, Srinivasan Parthasarathy, and Amit P Sheth. On understanding the divergence of online social group discussion. *ICWSM*, 2014.

[256] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 1999.

[257] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In *Advances in Neural Information Processing Systems*, pages 3239–3249, 2017.

[258] Suman Ravuri and Andreas Stoicke. A comparative study of neural network models for lexical intent classification. In *IEEE ASRU*, 2015.

[259] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *LREC Workshop on New Challenges for NLP Frameworks*, 2010.

[260] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2010.

[261] S. Riezler and J. Maxwell III. On some pitfalls in automatic evaluation and significance testing for mt. In *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.

[262] Gene I Rochlin. Mind the gap: The growing distance between institutional and technical capabilities in organizations performing critical operations. In *International Conference on Intelligence and Security Informatics*, pages 349–358. Springer, 2004.

[263] Matthew D Rocklage and Russell H Fazio. The evaluative lexicon: Adjective use as a means of assessing and distinguishing attitude valence, extremity, and emotionality. *Journal of Experimental Social Psychology*, 56:214–227, 2015.

[264] Matthew D Rocklage, Derek D Rucker, and Loran F Nordgren. The evaluative lexicon 2.0: The measurement of emotionality, extremity, and valence in language. *Behavior research methods*, 50(4):1327–1344, 2018.

[265] Daniel M Romero, Wojciech Galuba, Sitaram Asur, and Bernardo A Huberman. Influence and passivity in social media. In *ECML PKDD*. 2011.

[266] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, 2007.

[267] Abram Rosenblatt and Jeff Greenberg. Examining the world of the depressed: Do depressed people prefer others who are depressed? *Journal of Personality and Social Psychology*, 1991.

[268] Dan Roth and Wen-tau Yih. Integer linear programming inference for conditional random fields. In *Proceedings of ICML*, 2005.

[269] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 1987.

[270] Stephanie Rude, Eva-Maria Gortner, and James Pennebaker. Language use of depressed and depression-vulnerable college students. *Cognition & Emotion*, 2004.

[271] Itsumi Saito, Jun Suzuki, Kyosuke Nishida, Kugatsu Sadamitsu, Satoshi Kobashikawa, Ryo Masumura, Yuji Matsumoto, and Junji Tomita. Improving neural text normalization with data augmentation at character-and morphological levels. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 257–262, 2017.

[272] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171, 2016.

[273] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial Intelligence and Statistics*, 2009.

[274] R. Sampson, S. Raudenbush, and F. Earls. Neighborhoods and violent crime: A multilevel study of collective efficacy. *Science*, 1997.

[275] Cicero D Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1818–1826, 2014.

[276] Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*, 2015.

[277] Venu Satuluri and Srinivasan Parthasarathy. Bayesian locality sensitive hashing for fast similarity search. *VLDB Endowment*, 2012.

[278] Daniel Scanfeld, Vanessa Scanfeld, and Elaine L Larson. Dissemination of health information through social networks: Twitter and antibiotics. *American Journal of Infection Control*, 2010.

[279] M. Schlichtkrull and H.M. Alonso. Msejrku at semeval-2016 task 14: Taxonomy enrichment by evidence ranking. In *SemEval*, 2016.

[280] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[281] John R Searle. *Speech acts: An essay in the philosophy of language*. Cambridge university press, 1969.

[282] A. See, P. Liu, and C. Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint 1704.04368*, 2017.

[283] Sohil Atul Shah and Vladlen Koltun. Robust continuous clustering. *PNAS*, 2017.

[284] Gautam Kishore Shahi and Durgesh Nandini. Fakecovid–a multilingual cross-domain fact check news dataset for covid-19. *arXiv preprint arXiv:2006.11343*, 2020.

[285] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, and Y. Liu. Combating fake news: A survey on identification & mitigation techniques. *ACM TIST*, 2019.

[286] Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. A conditional variational framework for dialog generation. *arXiv preprint arXiv:1705.00316*, 2017.

[287] P. Shiralkar, A. Flammini, F. Menczer, and G. Ciampaglia. Finding streams in knowledge graphs to support fact checking. In *IEEE ICDM*, 2017.

[288] Prashanth Gurunath Shivakumar, Mu Yang, and Panayiotis Georgiou. Spoken language intent detection using confusion2vec. *arXiv preprint arXiv:1904.03576*, 2019.

[289] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu. defend: Explainable fake news detection. In *ACM SIGKDD*, 2019.

[290] Lei Shu, Hu Xu, and Bing Liu. Doc: Deep open classification of text documents. *arXiv preprint arXiv:1709.08716*, 2017.

[291] Lei Shu, Hu Xu, and Bing Liu. Unseen class discovery in open-world classification. *arXiv preprint arXiv:1801.05609*, 2018.

[292] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, et al. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11):1251, 2007.

[293] Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, DTIC Document, 1986.

[294] R. Snow, D. Jurafsky, and A.Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *NIPS*, 2005.

[295] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.

[296] L. Song, Y. Zhang, Z. Wang, and D. Gildea. N-ary relation extraction using graph state lstm. *arXiv preprint 1808.09101*, 2018.

[297] Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

[298] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014.

[299] Nitish Srivastava and Ruslan R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*, 2012.

[300] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[301] D. Stammbach and G. Neumann. Team domlin: Exploiting evidence enhancement for the fever shared task. In *FEVER Workshop*, 2019.

[302] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *JMLR*, 2002.

[303] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-informed self-attention for semantic role labeling. *arXiv preprint arXiv:1804.08199*, 2018.

[304] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *ACM World Wide Web*, 2007.

[305] A. Suhr, M. Lewis, J. Yeh, and Y. Artzi. A corpus of natural language for visual reasoning. In *ACL*, 2017.

[306] Asuka Sumida and Kentaro Torisawa. Hacking wikipedia for hyponymy relation acquisition. In *IJCNLP*, 2008.

[307] J. Sun, D. Ajwani, P. K. Nicholson, A. Sala, and S. Parthasarathy. Breaking cycles in noisy hierarchies. In *ACM Web Science*, 2017.

[308] Jiankai Sun, Deepak Ajwani, Patrick K. Nicholson, Alessandra Sala, and Srinivasan Parthasarathy. Breaking cycles in noisy hierarchies. In *ACM WebSci*, 2017.

[309] Ming Sun, Aasish Pappu, Yun-Nung Chen, and Alexander I Rudnicky. Weakly supervised user intent detection for multi-domain dialogues. In *IEEE SLT*, 2016.

[310] Osma Suominen and Eero Hyvönen. Improving the quality of skos vocabularies with skosify. In *Knowledge Engineering and Knowledge Management*, 2012.

[311] I. Sutskever, O. Vinyals, and Q. Le. Sequence to sequence learning with neural networks. *NeurIPS*, 2014.

[312] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[313] Piotr Sztompka. Trust: A sociological theory. In *Cambridge University Press*, 1999.

[314] Henri Tajfel. Social identity and intergroup relations. In *Cambridge University Press*, 2010.

[315] Liling Tan, Rohit Gupta, and Josef van Genabith. Usaar-wlv: Hypernym generation with deep neural nets. In *SemEval@ NAACL-HLT*, 2015.

[316] Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. Deep semantic role labeling with self-attention. *arXiv preprint arXiv:1712.01586*, 2017.

[317] Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. Deep semantic role labeling with self-attention. In *AAAI Conference on Artificial Intelligence*, 2018.

[318] Yuzuru Tanahashi and Kwan-Liu Ma. Design considerations for optimizing storyline visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 2012.

[319] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. Exploiting homophily effect for trust prediction. In *WSDM*, 2013.

[320] Jiliang Tang, Huiji Gao, and Huan Liu. mtrust: discerning multi-faceted trust in a connected world. In *WSDM*, 2012.

[321] Jiliang Tang and Huan Liu. *Trust in Social Media: Synthesis Lectures on Information Security, Privacy, and Trust*. Morgan & Claypool Publishers, 2015.

[322] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1972.

[323] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *JASIST*, 2012.

[324] J. Thorne and A. Vlachos. Automated fact checking: Task formulations, methods and future directions. *arXiv preprint 1806.07687*, 2018.

[325] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, and A. Mittal. The fact extraction and verification (fever) shared task. *arXiv preprint 1811.10971*, 2018.

[326] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, and A. Mittal. The fever2. 0 shared task. In *Workshop on Fact Extraction and VERification (FEVER)*, 2019.

[327] A. Toral and M. Monachini. Named entity wordnet. In *LREC*, 2008.

[328] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080, 2016.

[329] Luu Tuan, Yi Tay, Siu Hui, and See Ng. Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016.

[330] Dilek Hakkani Tur, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, 2016.

[331] Gokhan Tur, Dilek Hakkani Tür, Larry Heck, and Sarangarajan Parthasarathy. Sentence simplification for spoken language understanding. In *ICASSP*, 2011.

[332] Thomas W Valente. *Social networks and health: Models, methods, and applications*. 2010.

[333] G. Varelas, E. Voutsakis, P. Raftopoulou, E.G.M. Petrakis, and E.E. Milios. Semantic similarity methods in wordnet and their application to information retrieval on the web. In *ACM WIDM*, 2005.

[334] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, et al. Attention is all you need. In *NeurIPS*, 2017.

[335] Nikhita Vedula, Rahul Gupta, Aman Alok, and Mukund Sridhar. Automatic discovery of novel intents & domains from text utterances. *arXiv preprint arXiv:2006.01208*, 2020.

[336] Nikhita Vedula, Nedim Lipka, Pranav Maneriker, and Srinivasan Parthasarathy. Open intent extraction from natural language interactions. In *The Web Conference (WWW)*, 2020.

[337] Nikhita Vedula, Pranav Maneriker, and Srinivasan Parthasarathy. Bolt-k: Bootstrapping ontology learning via transfer of knowledge. In *The Web Conference (WWW)*, 2019.

[338] Nikhita Vedula, Pranav Maneriker, and Srinivasan Parthasarathy. Bolt-k: Bootstrapping ontology learning via transfer of knowledge. In *The World Wide Web Conference*, pages 1897–1908, 2019.

[339] Nikhita Vedula, Patrick K. Nicholson, Deepak Ajwani, Sourav Dutta, Alessandra Sala, and Srinivasan Parthasarathy. Enriching taxonomies with functional domain knowledge. In *ACM Conference on Research & Development in Information Retrieval (SIGIR)*, 2018.

[340] Nikhita Vedula, Patrick K Nicholson, Deepak Ajwani, Sourav Dutta, Alessandra Sala, and Srinivasan Parthasarathy. Enriching taxonomies with functional domain knowledge. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 745–754. ACM, 2018.

[341] Nikhita Vedula and Srinivasan Parthasarathy. Emotional and linguistic cues of depression from social media. In *ACM Digital Health (DH)*, 2017.

[342] Nikhita Vedula and Srinivasan Parthasarathy. Face-keg: Fact checking explained using knowledge graphs. In *In Review*, 2020.

[343] Nikhita Vedula, Srinivasan Parthasarathy, and Valerie L. Shalin. Predicting trust relations among users in a social network: On the role of influence, cohesion and valence. In *WISDOM workshop at SIGKDD*, 2016.

[344] Nikhita Vedula, Srinivasan Parthasarathy, and Valerie L. Shalin. Predicting trust relations within a social network: A case study on emergency response. In *ACM Web Science Conference (WebSci'17)*, 2017.

[345] Nikhita Vedula, Wei Sun, Hyunhwan Lee, Harsh Gupta, Mitsunori Ogihara, Joseph Johnson, Gang Ren, and Srinivasan Parthasarathy. Multimodal content analysis for

effective advertisements on youtube. In *IEEE International Conference on Data Mining (ICDM)*, 2017.

[346] Nikhita Vedula, Wei Sun, Hyunhwan Lee, Harsh Gupta, Mitsunori Ogihara, Joseph Johnson, Gang Ren, and Srinivasan Parthasarathy. Multimodal content analysis for effective advertisements on youtube. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1123–1128. IEEE, 2017.

[347] Nikhita Vedula, Wei Sun, Hyunhwan Lee, Harsh Gupta, Mitsunori Ogihara, Joseph Johnson, Gang Ren, and Srinivasan Parthasarathy. Multimodal content fanalysis for effective advertisements on youtube. *arXiv:1709.03946 (extended version of ICDM paper)*, 2017.

[348] Paola Velardi, Stefano Faralli, and Roberto Navigli. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3):665–707, 2013.

[349] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint 1710.10903*, 2017.

[350] Subhashini Venugopalan, Lisa Anne Hendricks, Raymond Mooney, and Kate Saenko. Improving lstm-based video description with linguistic knowledge mined from text. *arXiv preprint arXiv:1604.01729*, 2016.

[351] Sarah Vieweg, Amanda L Hughes, Kate Starbird, and Leysia Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *ACM SIGCHI*, 2010.

[352] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *JMLR*, 2010.

[353] Piek Vossen. Extending, trimming and fusing wordnet for technical documents. ACL, 2001.

[354] VG Vydiswaran, ChengXiang Zhai, and Dan Roth. Content-driven trust propagation framework. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 974–982. ACM, 2011.

[355] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*, 2019.

[356] Bo Wang, Jinqiao Wang, and Hanqing Lu. Exploiting content relevance and social relevance for personalized ad recommendation on internet tv. *ACM TOMM*, 2013.

[357] Chi Wang, Marina Danilevsky, Nihit Desai, Yinan Zhang, Phuong Nguyen, Thrivikrama Taula, and Jiawei Han. A phrase mining framework for recursive construction of a topical hierarchy. In *ACM SIGKDD*, 2013.

[358] Jinpeng Wang, Gao Cong, Wayne Xin Zhao, and Xiaoming Li. Mining user intents in twitter: A semi-supervised approach to inferring intent categories for tweets. In *AAAI*, 2015.

[359] Yu Wang, Yilin Shen, and Hongxia Jin. A bi-model based rnn semantic frame parsing model for intent detection and slot filling. *arXiv preprint arXiv:1812.10235*, 2018.

[360] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 2014.

[361] Walter Weintraub. *Verbal behavior: Adaptation and psychopathology*. 1981.

[362] Laura Wendlandt, Rada Mihalcea, Ryan L Boyd, and James W Pennebaker. Multi-modal analysis and prediction of latent user dimensions. In *International Conference on Social Informatics*, pages 323–340. Springer, 2017.

[363] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*, 2010.

[364] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*. 2012.

[365] Dominic Widdows. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *NAACL-HLT*, 2003.

[366] John Wieting, Jonathan Mallinson, and Kevin Gimpel. Learning paraphrastic sentence embeddings from back-translated bitext. *arXiv preprint arXiv:1706.01847*, 2017.

[367] Baoning Wu, Vinay Goel, and Brian D Davison. Topical trustrank: Using topicality to combat web spam. In *WWW*, 2006.

[368] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *ACL*, 1994.

[369] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. Zero-shot user intent detection via capsule neural networks. *arXiv preprint arXiv:1809.00385*, 2018.

[370] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *AAAI Conference on Artificial Intelligence*, pages 2659–2665, 2016.

[371] Hu Xu, Bing Liu, Lei Shu, and P Yu. Open-world learning and application to product classification. In *WWW*, 2019.

[372] Puyang Xu and Ruhi Sarikaya. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *ASRU*, 2013.

[373] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *SIGIR*, 2003.

[374] I. Yamada, J. Oh, C. Hashimoto, K. Torisawa, Jun'ichi K., S. De Saeger, and T. Kawada. Extending wordnet with hypernyms and siblings acquired from wikipedia. In *IJCNLP*, 2011.

[375] I. Yamada, K. Torisawa, J. Kazama, K. Kuroda, M. Murata, S. De Saeger, F. Bond, and A. Sumida. Hypernym discovery based on distributional similarity and hierarchical structures. In *EMNLP*, 2009.

[376] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[377] Bo Yang, Tao Mei, Xian-Sheng Hua, Linjun Yang, Shi-Qiang Yang, and Mingjing Li. Online video recommendation based on multimodal fusion and relevance feedback. In *ACM CIVR*, 2007.

[378] F. Yang, S. K. Pentyala, S. Mohseni, M. Du, H. Yuan, R. Linder, E. D Ragan, S. Ji, and X. Hu. Xfake: explainable fake news detector with visualizations. In *The World Wide Web Conference (WWW)*, 2019.

[379] Hui Yang and Jamie Callan. A metric-based framework for automatic taxonomy induction. In *ACL*, 2009.

[380] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv:1603.08861*, 2016.

[381] Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*, 2017.

[382] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[383] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.

[384] S. Yao, T. Wang, and X. Wan. Heterogeneous graph transformer for graph-to-sequence learning. In *ACL*, 2020.

[385] Jinfeng Yi, Lijun Zhang, Tianbao Yang, Wei Liu, and Jun Wang. An efficient semi-supervised clustering algorithm with sequential constraints. In *ACM SIGKDD*, 2015.

[386] T. Yoneda, J. Mitchell, J. Welbl, P. Stenetorp, and S. Riedel. Ucl machine reading group: Four factor framework for fact finding (hexaf). In *FEVER workshop*, 2018.

[387] Jianxing Yu, Zheng-Jun Zha, Meng Wang, Kai Wang, and Tat-Seng Chua. Domain-assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 140–150. Association for Computational Linguistics, 2011.

[388] Qian Yu and Wai Lam. Product question intent detection using indicative clause attention and adversarial learning. In *Proceedings of ACM SIGIR*, 2018.

[389] Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. Learning term embeddings for hypernymy identification. In *IJCAI*, 2015.

[390] Jianbo Yuan, Han Guo, Zhiwei Jin, Hongxia Jin, Xianchao Zhang, and Jiebo Luo. One-shot learning for fine-grained relation extraction via convolutional siamese neural network. In *2017 IEEE International Conference on Big Data*, pages 2194–2199. IEEE, 2017.

[391] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim. Graph transformer networks. In *NeurIPS*, 2019.

[392] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, 2015.

[393] Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Incorporating relation paths in neural relation extraction. *arXiv preprint arXiv:1609.07479*, 2016.

[394] Huifeng Zhang, Su Zhu, Shuai Fan, and Kai Yu. Joint spoken language understanding and domain adaptive language modeling. In *International Conference on Intelligent Science and Big Data Engineering*, 2018.

[395] Jing Zhang, Wanqing Li, and Philip Ogunbona. Transfer learning for cross-dataset recognition: a survey. *arXiv preprint arXiv:1705.04396*, 2017.

[396] Xiang Zhang and Yann LeCun. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*, 2015.

[397] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

[398] Xiaodong Zhang and Houfeng Wang. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*, 2016.

[399] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, 2018.

[400] J. Zhou, X. Han, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Gear: Graph-based evidence aggregating and reasoning for fact verification. *arXiv preprint 1908.01843*, 2019.

[401] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212, 2016.

[402] Cai-Nicolas Ziegler and Jennifer Golbeck. Investigating interactions of trust and interest similarity. *Decision support systems*, 2007.