ALTA 2020

Proceedings of the 18th Workshop of the Australasian Language Technology Association

14–15 January, 2021
Virtual Workshop

Sponsors

Platinum

Australian Government

Department of Defence

Silver

SINTELIX

Google

Introduction

Welcome to the 18th edition of the Annual Workshop of the Australasian Language Technology Association (ALTA 2020). The purpose of ALTA is to promote language technology research and development in Australia and New Zealand. Every year ALTA hosts a workshop which is the key local forum for disseminating research in Natural Language Processing and Computational Linguistics, with presentations and posters from students, industry, and academic researchers. This year ALTA is hosted as a virtual workshop, due to the COVID-19 pandemic.

In total we received 25 paper submissions and we accepted 8 long papers and 8 short papers to appear in the workshop, as well as 3 extended abstracts. Of all submissions, 19 were first-authored by students. We had submissions from a total of five countries: Australia, New Zealand, Japan, Sri Lanka and United States. We are extremely grateful to the Programme Committee members for their time and their detailed and helpful comments and reviews. This year we had committee members from all over the globe including Australia, New Zealand, Japan, Sweden, Switzerland, United States and United Arab Emirates.

Overall, there will be two oral presentation sessions and two virtual poster sessions. We also ran a shared task in detection of human behaviour organised by Diego Mollá-Aliod (University of Macquarie), In addition, for the first time ALTA will have a Doctoral Consortium, with a single session for final year PhD students and recent PhD graduates to present their work to help young researchers to gain visibility to their prospective employers. Finally, the workshop will feature keynotes from Kendra Vant (Xero) and Andrew Perfors (University of Melbourne), following a tradition of bringing speakers from both academia and industry.

ALTA 2020 is very grateful for the financial support generously offered by our sponsors. Without their contribution, the running of these events to bring together the NLP community of the Australasian region would have been a challenge. We would like to express sincere gratitude to our sponsors.

We very much hope that you will have an enjoyable and inspiring time at ALTA 2020!

Maria Kim and Daniel Beck

Organisers:

Program Co-Chair: Maria Kim, Defence Science and Technology Group
Program Co-Chair: Daniel Beck, The University of Melbourne
Program Advisor: Meladel Mistica, The University of Melbourne

Program Committee:

Timothy Baldwin, Jennifer Biggs, Benjamin Boerschinger, Wray Buntine, Lawrence Cavedon, Trevor Cohn, Xiang Dai, Lea Frermann, Gholamreza Haffari, Hamed Hassanzadeh, Brian Hur, Nitin Indurkhya, Antonio Jimeno-Yepes, Sarvnaz Karimi, Sunghwan Mac Kim, Alistair Knott, Yitong Li, David Martinez, Nitika Mathur, Diego Mollá-Aliod, Scott Nowson, Cecile Paris, Lizhen Qu, Afshin Rahimi, Abeed Sarker, Andrea Schalley, Rolf Schwitter, Laurianne Sitbon, Kristin Stock, Hanna Suominen, Karin Verspoor, Stephen Wan, Michael Witbrock, Hiyori Yoshikawa, Xiuzhen Zhang

Invited Speakers:

Kendra Vant, Xero
Andrew Perfors, The University of Melbourne

PROGRAMME - All times are in AEDT (Melbourne/Sydney).

14th December (Thursday) - Day 1

10:30 - 11:00 Opening Remakrs

11:00 - 12:00 Keynote: Kendra Vant (Xero)
Commercial machine learning at scale - the joys and the pitfalls

12:00 - 13:00 Lunch & Virtual Networking

13:00 - 14:00 Session 1 – Long Papers 1 (Session Chair: Cécile Paris)

Domain Adaptative Causality Encoder
Farhad Moghimifar, Gholamreza Haffari and Mahsa Baktashmotlagh
Automated Detection of Cyberbullying Against Women and Immigrants and Cross-domain Adaptability
Thushari Atapattu, Mahen Herath, Georgia Zhang and Katrina Falkner
The Influence of Background Data Size on the Performance of a Score-based Likelihood Ratio System: A Case of Forensic Text Comparison
Shunichi Ishihara
Feature-Based Forensic Text Comparison Using a Poisson Model for Likelihood Ratio Estimation
Michael Carne and Shunichi Ishihara

14:00 - 15:30 Afternoon Break & Poster Session (includes papers from Session 1)

Learning Causal Bayesian Networks from Text
Farhad Moghimifar, Afshin Rahimi, Mahsa Baktashmotlagh and Xue Li
Information Extraction from Legal Documents: A Study in the Context of Common Law Court Judgements
Meladel Mistica, Geordie Z. Zhang, Hui Chia, Kabir Manandhar Shrestha, Rohit Kumar Gupta, Saket Khandelwal, Jeannie Paterson, Timothy Baldwin and Daniel Beck
Benchmarking of Transformer-Based Pre-Trained Models on Social Media Text Classification Datasets
Yuting Guo, Xiangjue Dong, Mohammed Ali Al-Garadi, Abeed Sarker, Cecile Paris and Diego Mollá Aliod
Pandemic Literature Search: Finding Information on COVID-19
Vincent Nguyen, Maciek Rybinski, Sarvnaz Karimi and Zhenchang Xing
Leveraging Discourse Rewards for Document-Level Neural Machine Translation
Inigo Jauregi Unanue, Nazanin Esmaili, Gholamreza Haffari and Massimo Piccardi
The Open Domain Interviewing Agent

Ming-Bin Chen and Michael Witbrock

Cost-effective Selection of Pretraining Data: A Case Study of Pretraining BERT on Social Media

Xiang Dai, Sarvnaz Karimi, Ben Hachey and Cécile Paris

15:30 - 16:15 Session 2 – Doctoral Consortium (Session Chair: Stephen Wan)

Recognizing Biomedical Names: Challenges and Solutions

Xiang Dai

Ngana Wubulku Junkurr-Jiku Balkaway-Ka: The Intergenerational Co-Design of a Tangible Technology to Keep Active Use of the Kuku Yalanji Aboriginal Language Strong

Jennyfer Lawrence Taylor

Automatic Generation of Security-Centric Description for Cyber Threats

Tingmin Wu

16:15 End of ALTA 2020 Day 1

15th January (Friday) Day 2

11:00 - 12:00 Keynote: Andrew Perfors (The University of Melbourne)
TBD

12:00 - 13:00 Lunch & Virtual Networking

13:00 - 14:00 Session 3 – Long Papers 2 (Session Chair: Trevor Cohn)

Modelling Verbal Morphology in Nen

Saliha Muradoglu, Nicholas Evans and Ekaterina Vylomova

An Automatic Vowel Space Generator for Language Learner Pronunciation Acquisition and Correction

Xinyuan Chao, Charbel El-Khaissi, Nicholas Kuo, Priscilla Kan John and Hanna Suominen

ABSA-Bench: Towards the Unified Evaluation of Aspect-based Sentiment Analysis Research

Abhishek Das and Wei Emma Zhang

A machine-learning based model to identify PhD-level skills in job ads

Li'An Chen, Inger Mewburn and Hanna Suonimen

14:00 - 15:30 Afternoon Break & Poster Session (includes papers from Sessions 2 & 3)

Transformer Semantic Parsing

Gabriela Ferraro and Hanna Suominen

Convolutional and Recurrent Neural Networks for Spoken Emotion Recognition

Aaron Keesing, Ian Watson and Michael Witbrock

Popularity Prediction of Online Petitions using a Multimodal DeepRegression Model
Kotaro Kitayama, Shivashankar Subramanian and Timothy Baldwin
Exploring Looping Effects in RNN-based Architectures
Andrei Shcherbakov, Saliha Muradoglu and Ekaterina Vylomova

15:30 - 17:00 Session 4 – Shared Task, AGM, Best Paper and Closing

Shared Task (Chair: Diego Mollá-Aliod)
AGM (Chair: Sarvnaz Karimi)
Best Paper Awards (Chair: Maria Kim)
Closing (Chair: Daniel Beck)

17:00 End of ALTA 2020 Day 2

Table of Contents

Long Papers

Short Papers

Shared Task (Not Peer Reviewed)

Domain Adaptative Causality Encoder

Farhad Moghimifar[1], **Gholamreza Haffari**[2] and **Mahsa Baktashmotlagh**[1]
[1]The School of ITEE, The University of Queensland, Australia
[2]Faculty of Information Technology, Monash University, Australia
{f.moghimifar,m.baktashmotlagh}@uq.edu.au
gholamreza.haffari@monash.edu,

Abstract

Automated discovery of causal relationships from text is a challenging task. Current approaches which are mainly based on the extraction of low-level relations among individual events are limited by the shortage of publicly available labelled data. Therefore, the resulting models perform poorly when applied to a distributionally different domain for which labelled data did not exist at the time of training. To overcome this limitation, in this paper, we leverage the characteristics of dependency trees and adversarial learning to address the tasks of adaptive causality identification and localisation. The term adaptive is used since the training and test data come from two distributionally different datasets, which to the best of our knowledge, this work is the first to address. Moreover, we present a new causality dataset, namely MEDCAUS[1], which integrates all types of causality in the text. Our experiments on four different benchmark causality datasets demonstrate the superiority of our approach over the existing baselines, by up to 7% improvement, on the tasks of identification and localisation of the causal relations from the text.

1 Introduction

Causality is the basis for reasoning and decision making. While human-beings use this psychological tool to choreograph their environment into a mental model to act accordingly (Pearl and Mackenzie, 2018), the inability to identify causal relationships is one of the drawbacks of current Artificial Intelligence systems (Lake et al., 2015). The projection of causal relations in natural language enables machines to develop a better understanding of the surrounding context and helps downstream tasks such as question answering (Hassanzadeh et al., 2019), text summarisation (Ning et al., 2018), and natural language inference (Roemmele et al., 2011).

The task of textual causality extraction can be divided into two main subtasks, causality identification and causality localisation. The former subtask focuses on identifying whether a sentence carries any causal information or not, which can be seen as classification problem. The objective of the latter subtask is to extract text spans related to cause and effect, subject to existence.

The automatic identification and localisation of causal relations in textual data is considered a non-trivial task (Dasgupta et al., 2018). Causal relations in text can be categorised as marked/unmarked and explicit/implicit (Blanco et al., 2008; Hendrickx et al., 2009). Marked causality refers to the case where a causal linguistic feature, such as "because of", is stated in the sentence. For example, in *"His OCD is because of genetic factors."*, *because of* is a causal marker, whereas in unmarked causality there is no such indicator. For instance, in *"Don't take these medications before driving. you might feel sleepy."* the cause and effect relationship is spread between two sentences without a marker. On the other hand, explicit causality refers to the case where both cause and effect are mentioned in text. However, in implicit causality, either cause or effect are directly mentioned in the text. In a more complex case called nested causality, multiple causal relations may exist in one sentence (e.g., *"Procaine can also cause allergic reactions causing individuals to have problems with breathing"*). All of these ambiguities contribute to the challenging nature of this task.

Traditional approaches to address the problem of causality extraction mainly relied on predefined linguistic patterns and rules to identify the existence of causal relations in a sentence (Mirza and Tonelli, 2016). More advanced approaches combined pattern-based methods with machine learning

[1]https://github.com/farhadmfar/ace

techniques (Zhao et al., 2018), and as such they require heavy manual feature engineering to perform reasonably. To overcome this problem, the recent approaches have adopted deep learning techniques to extract meaningful features from the text (Liang et al., 2019; Martínez-Cámara et al., 2017).

However, all the aformentioned approaches suffer from the problem of domain shift, where there is a distribution difference between the training and the test data. More specifically, the existing approaches perform poorly on the data from a new test domain (e.g. financial) which is contextually different from the training domain (e.g. medical).

To overcome the limitations of existing approaches on the tasks of causality identification and localisation, we propose a novel approach for domain-adaptive causality encoding which performs equally well when applied on the out-of-domain sentences. Our contribution is three-fold:

- To identify causal relationships and extract the corresponding causality information within a sentence, using graph convolutional networks, we propose a model which takes into account both syntactic and semantic dependency of words in a sentence. Extensive experimental results suggest that our proposed models for causality identification and localisation outperform the state-of-the-art results.

- We propose to use a gradient reversal approach to minimise the distribution shift between the training and test datasets. Our proposed adaptive approach improves the performance of the existing baselines by up to 7% on the tasks of adaptive causality identification.

- To fill the gap of the current causality datasets on encompassing different types of causality, we introduce MEDCAUS, a dataset of 15,000 labelled sentences, retrieved from medical articles. This dataset consists of sentences with labels of explicit, implicit, nested, and no causality.

2 Related Works

The projection of causal relation in textual data can be in various forms, depending on the type of causality. The categorisation mentioned in Section 1 can indicate the relation between pairs of events, phrases, concepts, named entities or a mixture of the aforementioned text spans (Hashimoto, 2019). Some works in the area have endeavoured to extract and present the textual information between concepts or events. Causal relations are a component of SemEval task (Hendrickx et al., 2009), but it involves a limited set of causal relations between pairs of nominals. Do et al. (2011) developed a framework based on combining semantic association and supervised causal discourse classification in order to identify causal relations between pairs of events. They expand the patterns in *pdtb* (Lin et al., 2009) using a self-training approach. Other methods (Riaz and Girju, 2014, 2013) leveraged linguistic features such as part-of-speech information, alongside with discourse markers, for identifying causal relations between events. An et al. (2019) used the syntactic patterns and word vectors to develop an unsupervised method for constructing causal graphs. To expand the repository of causal syntactic patterns, Hidey and McKeown (2016) built a parallel corpus between English Wikipedia and Simple Wikipedia, where the same causal relation might be in different syntactic markers in two parallel sentences. A supervised method was adapted by Mirza and Tonelli (2016) using lexical, semantic, and syntactic features within a sentence to address this task.

Using Hidey and McKeown (2016)'s method, Martínez-Cámara et al. (2017) created a set of labelled sentences, assuming all of the sentences include a causal relation, and presented a neural model based on LSTM to identify causality. Dasgupta et al. (2018) collected a dataset and developed a model using BiLSTM for extracting causal relation within a sentence[2]. Other approaches in event prediction applied Granger causality (Granger, 1988) to identify causal relations in time series of events (Kang et al., 2017). Rojas-Carulla et al. (2017) defined a proxy variable, which may carry some information about cause and effect, to identify causal relationship between static entities. Zhao et al. (2017) developed a causality network embedding for event prediction. De Silva et al. (2017) proposed a convolutional neural network model for identifying causality. Liang et al. (2019) also deployed a self-attentive neural model to address the task of causality identification, however, the extraction of causal information is not addressed by their model.

In more recent works, a dataset of counterfactual

[2]The source code and dataset are not publicly available.

sentences was released, as a part of SemEval2020 Task5 (Yang et al., 2020). The aim of this task is to identify and tag the existing counterfactual part of the sentence. Some works have attended to address this task using different deep learning architectures (Patil and Baths, 2020; Abi Akl et al., 2020). While counterfactuals are usually represented in form of a causal relation, this dataset does not cover different forms of textual causality.

As opposed to the aforementioned models, we propose a unified neural model for addressing both tasks of identifying and localising causality from a sentence. Our method leverages both syntactic and semantic relations within a sentence, and adapts to out-of-domain sentences.

3 Our Approach

In this section, we first describe the architecture of causality extractor, which uses graph convolutional networks (GCN) at its core. We then present how we make use of the adversarial learning strategy for adapting the model to new domains. Figure 1 illustrates the high level overview of our approach.

3.1 Graphical Causality Encoder (GCE)

Given a sentence $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, where $\boldsymbol{x}_i$ is the vector representation of the i-th token of the sentence, the goal of our model is two-fold: identifying whether or not the causal relation exists, and locating the position of cause and effect in the given sentence.

The core part of our causal identification model consists of an L-layer graph convolutional network (GCN) which takes as input the dependency tree of a sentence, obtained through Stanford CoreNLP (Manning et al., 2014). The dependency tree can be represented with an $n \times n$ adjacency matrix $\boldsymbol{A}$, where n is the number of nodes in the graph. In the adjacency matrix, $A_{ji} = A_{ij} = 1$ if an edge connects i to j, and zero otherwise. Given $\boldsymbol{h}_i^{(l-1)}$ as the representation of the node i at layer $l-1$, GCN updates the node representation at layer l as follows (Zhang et al., 2018; Kipf and Welling, 2016):

$$\boldsymbol{h}_i^{(l)} = \boldsymbol{f}^{\text{actv}}\Big(\sum_{j=1}^{n} \tilde{A}_{ij}\mathbf{W}^{(l)}\boldsymbol{h}_j^{(l-1)}/d_i + \boldsymbol{b}^{(l)}\Big) \quad (1)$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$, with $\boldsymbol{I}$ the identity matrix, $\boldsymbol{f}^{\text{actv}}$ an activation function (i.e., element-wise RELU), $\boldsymbol{b}^{(l)}$ the bias vector, $\mathbf{W}^{(l)}$ the weight matrix, and $d_i = \sum_{j=1}^{n} \tilde{A}_{ij}$ the degree of node i.

This formation captures the hidden embeddings of each token in a sentence with respect to its neighbours with maximum distance of L, with L the number of GCN layers. To take the words order and disambiguity into account and make the model less prone to errors from the dependency relations' results, we feed the word vectors into a bi-directional long short-term memory (BiLSTM) network. The output of the BiLSTM $\boldsymbol{h}_i^{(0)}$ is then used in the GCN, as illustrated in Equation 1. Hence, after applying the BiLSTM and GCN, each sentence is represented as:

$$\boldsymbol{h}_{\text{GCN}}(\boldsymbol{X}) = \boldsymbol{f}^{\text{pool}}(\text{GCN}(\text{BiLSTM}(\boldsymbol{X})) \quad (2)$$

where $\boldsymbol{f}^{\text{pool}} : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^d$ is a pooling function generating the representations for the n tokens of the sentence. The final sentence representation is obtained by a feed forward network (FFNN) whose input is the concatenation of $\boldsymbol{h}_{\text{GCN}}(\boldsymbol{X})$ and $\boldsymbol{h}_{\text{BiLSTM}}(\boldsymbol{X})$. Note that $\boldsymbol{h}_{\text{BiLSTM}}(\boldsymbol{X})$ is the contextualised representation of the sentence from BiLSTM which is constructed by concatenating the leftmost and rightmost hidden states:

$$\boldsymbol{f}_{\boldsymbol{\theta}_{\text{enc}}}(\boldsymbol{X}) = \text{FFNN}([\boldsymbol{h}_{\text{GCN}}(\boldsymbol{X}); \boldsymbol{h}_{\text{BiLSTM}}(\boldsymbol{X})]) \quad (3)$$

where $\boldsymbol{\theta}_{\text{enc}}$ contains the collection of parameters of the GCN, BiLSTM, and the feed-forward network. This representation is then used to address the two main sub-tasks:

- For Task1, which is identifying causal relation within a sentence, we use this representation to get the probability of output classes,

$$P_{\boldsymbol{\theta}_{\text{class}}}(\text{causality}|\boldsymbol{X}) = \sigma(\boldsymbol{f}_{\boldsymbol{\theta}_{\text{enc}}}(\boldsymbol{X}).\mathbf{W}_{\text{class}} + b_{\text{class}}) \quad (4)$$

  where $\boldsymbol{\theta}_{class} := \{\mathbf{W}_{\text{class}}, b_{\text{class}}\}$ contains the classifier's parameters, and σ is the sigmoid function.

- For Task2, locating cause and effect in a sentence, we use this representation to obtain the probability of the corresponding tag for each token. Since there are strong dependencies across tags, by adopting conditional random fields, we model the tagging decision jointly, with respect to surrounding tags. Consider $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]$ the sequence of tag predictions. The score corresponding to this sequence is defined as:

$$\boldsymbol{s}(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=0}^{n} \boldsymbol{f}_{\boldsymbol{\theta}_{\text{enc}}}(\boldsymbol{X})_{i,\boldsymbol{y}_i} + \sum_{i=0}^{n} T_{\boldsymbol{y}_i,\boldsymbol{y}_{i+1}} \quad (5)$$

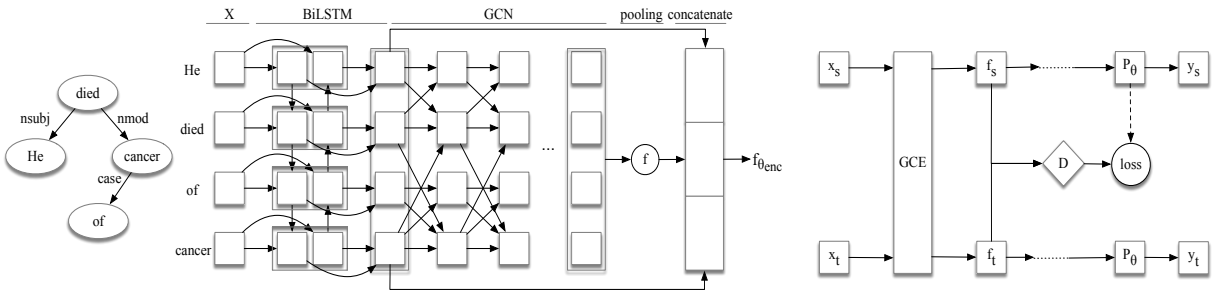


Figure 1: (Left) The dependency tree of the sentence *"He died of cancer."*. (Middle) Architecture of our proposed models for Causality Identifier (GCE), which uses the retrieved dependency tree as its core. (Right) The architecture of our proposed Adaptive Causality Encoder (ACE), which uses the structure of GCE as feature extractor.

where T is a square matrix with its size corresponding to the number of distinct tags. $T_{i,j}$ is representative of the score related to shifting from tag *i* to tag *j*. The probability of the tag sequence $\boldsymbol{Y}$ given $\boldsymbol{X}$ is then defined as ($\boldsymbol{Y_X}$ which denotes all possible sequences of tags for $\boldsymbol{X}$):

$$P_{\boldsymbol{\theta}_{\text{seq}}}(\boldsymbol{Y}|\boldsymbol{X}) = \boldsymbol{s}(\boldsymbol{X}, \boldsymbol{Y}) - log(\sum_{\boldsymbol{y} \in \boldsymbol{Y_X}} e^{s(\boldsymbol{X}, \boldsymbol{y})}) \quad (6)$$

Here, $\boldsymbol{\theta}_{\text{seq}}$ contains the sequence tagger's parameters.

3.2 Adaptive Causality Encoder (ACE)

In this section, we represent a domain adversarial approach to adaptive causality identification and localisation. In unsupervised domain adaptation, we are given a source labelled data $D_s = \{(\boldsymbol{X}_s^{(i)}, Y_s^{(i)})\}_{i=1}^{n_s}$ and unlabelled target data $D_t = \{\boldsymbol{X}_t^{(j)}\}_{j=1}^{n_t}$. Our aim is to reduce the distributional shift between the two domains, and predict the labels of the target domain. Inspired by (Ganin et al., 2016; Long et al., 2018), we make use of an adversarial learning strategy, where the domain discriminator is trained to distinguish the source and domains, while the feature representation is trained to confuse the domain discriminator.

More formally, let us consider the following domain classifier,

$$P_{\boldsymbol{\theta}_{\text{dom}}}(\text{source}|\boldsymbol{X}) = \sigma(\boldsymbol{f}_{\boldsymbol{\theta}_{\text{enc}}}(\boldsymbol{X}).\mathbf{W}_{\text{dom}} + b_{\text{dom}}) \quad (7)$$

where $\boldsymbol{\theta}_{dom} := \{\mathbf{W}_{\text{dom}}, b_{\text{dom}}\}$ is the domain classifier's parameters. Our domain adversarial training objective is defined as,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dom}}, \boldsymbol{\theta}_{\text{task}}) := & \sum_{(\boldsymbol{X},Y)\in D_s} \log P_{\boldsymbol{\theta}_{\text{class}}}(Y|\boldsymbol{f}_{\boldsymbol{\theta}_{\text{enc}}}(\boldsymbol{X})) \\ & - \sum_{\boldsymbol{X}\in D_s} \log P_{\boldsymbol{\theta}_{\text{dom}}}(\text{source}|\boldsymbol{f}_{\boldsymbol{\theta}_{\text{enc}}}(\boldsymbol{X})) \\ & - \sum_{\boldsymbol{X}\in D_t} \log\big(1 - P_{\boldsymbol{\theta}_{\text{dom}}}(\text{source}|\boldsymbol{f}_{\boldsymbol{\theta}_{\text{enc}}}(\boldsymbol{X}))\big).\end{aligned}$$

The model parameters are then trained by,

$$\arg\max_{\boldsymbol{\theta}_{\text{task}}} \max_{\boldsymbol{\theta}_{\text{enc}}} \min_{\boldsymbol{\theta}_{\text{dom}}} \mathcal{L}(\boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dom}}, \boldsymbol{\theta}_{\text{task}})$$

where minimization over $\boldsymbol{\theta}_{\text{dom}}$ strengthens the accuracy of the domain classifier, but maximizing over $\boldsymbol{\theta}_{\text{enc}}$ tries to confuse the domain classifier and strengthen the causality classifier.

3.3 Tagging Scheme

The objective of the task of causality localisation is to assign a label to each token in a sentence to locate the position of cause and effect. Cause and effect of a causal relation may span several tokens in a sentence. Therefore, the labels of a sentence usually are represented in the IOB-format (Inside, outside, and beginning). In this format, B-*label* indicates beginning of the span *label*, I-*label* shows a token inside the *label* but not the first token, and O-*label* represents the token as an outsider of *label*. However, inspired by (Ratinov and Roth, 2009) and (Dai et al., 2015), we use IOBES, an extended version of IOB, which also accounts for singleton labels and end of the label span token. Furthermore, to keep the tags consistent with the Equation 5, we add a *start* and *end* label to the set of tags.

4 Experiments

In this section, we first describe the datasets that have been used for the evaluation of our models, including our collected dataset. Then we present results of our proposed models on both (adaptive) causality identification and causality localisation.

4.1 Datasets

MEDCAUS We introduce our medical causality dataset with 15,000 sentences. The process of collection and annotation of the sentences was followed by the guideline of Hendrickx et al. (2009),

#causality classes	4
average sentence length	29.5
#explicit causality	9,092
#implicit causality	616
#nested causality	1,356
#non-causal	3,936
#Total sentences	15,000

Table 1: Statistics on MEDCAUS, our collected dataset.

including three main phases. In the first phase, sentences from medical articles of Wikipedia were randomly extracted. Using a wide variety of predefined causal connective words and patterns, we manually selected the sentences with potential causal relation and those without causal relation. In the second phase, the annotation instruction, multiple examples with different types of causal relation (i.e., explicit, implicit, nested, and non-causal) and different causal connective words were provided to the annotators. We asked four English-speaking graduate students to label the data accordingly. In the third phase, sentences with any disagreement that could not be resolved or were not clear in terms of causal relation were removed. To measure the level of agreement between our annotators, we give the same set of 1,000 sentences to the annotators. Using Fleiss Kappa measure (Fleiss and Cohen, 1973) ($\hbar$), the level of agreement between our annotators has been 0.71, showing the reliability of the annotations. Table 1 reports statistics about our collected dataset.

FinCausal The dataset, which is extracted from financial news provided by QWAM [3], includes different sets for both tasks of causality identification and localisation. For the former, it includes 22,058 sentences, and for the latter task, 1,750 sentences were provided[4].

SemEval-10 We use SemEval-10 Task 8, which has 1,003 sentences with causal relation. From other relations of this dataset, we randomly select 997 sentences, totalling 2,000 sentences. The sentences from this dataset are selected from a wide variety of domains, however, unlike MEDCAUS the causal relations are indicated only between pair nominals. This dataset was used for both causality identification and causality localisation task (Hendrickx et al., 2009).

BioCausal-Small The dataset is a part of larger dataset [5], consisting of 2,000 biomedical sentences from which 1,113 have causal relations. The sentences from this dataset have been collected from biomedical articles of PubMed [6]. Since this dataset only includes information about whether a sentence has causal relation (regardless of the position of the cause and effect), it has been used for causality identification (Kyriakakis et al., 2019).

4.2 Experimental Details

For both GCE and ACE, we use Stanford CoreNLP (Manning et al., 2014) to generate the dependency parsing tree for each sentence. We use the pre-trained 300-dimensional Glove vectors (Pennington et al., 2014) to initialise the embedding layer of our model. The hidden size for LSTM and the output feedforward layers is set to 100. We use the standard max pooling function for the pooling layer. Also, for all non-linearities in our model, we use Tanh function. A dropout ratio of $p = 0.5$ has been applied to all layers except for the last layer of GCN, for regularisation purposes.

For training of GCE, we split the data into train, development, and test set with the ratio of 60:20:20. For both models, we use batches of size 50. We train the model for 100 epochs, using Adamax optimiser. We use a decay rate of 0.9 if the $F1$ score of development set does not increase after each epoch. The reported results are micro-averaged precision, recall, and F1 score. All the hyperparameter and training settings were kept the same as reported above for other models for comparison. The original GCN and C-GCN model (Zhang et al., 2018), which have been used as baselines for experiment, use the Named Entity Recognition and Part of Speech Tagging embeddings of the related named entity as input to the model. Since identifying causal relation is not limited to named entities only, to be able to adjust baseline models to our experiment setup, we trained these model without the aforementioned embeddings.

4.3 Task1: Causality identification

In this section, we report the results on the task of identifying whether a sentence includes any causal relation or not. For this purpose, we use MEDCAUS and FinCausal to compare our GCE-based classifier (c.f. §3.1) with existing methods

[3] http://www.qwamci.com/
[4] http://wp.lancs.ac.uk/cfie/fincausal2020/
[5] The complete dataset is not publicly available.
[6] https://pubmed.ncbi.nlm.nih.gov

	MEDCAUS			FinCausal		
Model	P	R	F1	P	R	F1
P-Wiki (Hidey and McKeown, 2016)	74.4	74.4	74.4	54.0	54.0	54.0
bi-LSTM (Martínez-Cámara et al., 2017)	84.2	**97.8**	90.5	81.3	77.0	79.1
GCN (Zhang et al., 2018)	90.8	94.6	92.7	85	74.8	79.6
C-GCN (Zhang et al., 2018)	91.2	94.9	93.0	**86.1**	68.7	76.4
GCE	**92.5**	94.0	**93.2**	84.8	**83.3**	**84**

Table 2: Results of our proposed method on Task1, causality identification, compared with the baseline approaches on the MEDCAUS and FinCausal dataset.

	MEDCAUS			FinCausal		
Model	P	R	F1	P	R	F1
bi-LSTM-CRF (Martínez-Cámara et al., 2017)	**77.4**	69.9	73.4	**82.4**	65.0	72.7
GCN-CRF (Zhang et al., 2018)	31.9	46.8	37.9	66.1	55.5	60.3
C-GCN-CRF (Zhang et al., 2018)	72.5	**75.9**	74.1	76.3	68.8	72.3
S-LSTM-CRF (Lample et al., 2016)	58.6	64.0	61.2	61.5	29.7	40.0
ELMO-CRF (Peters et al., 2018)	48.5	78.9	60.1	71.8	61.3	66.1
GCE	76.3	73.6	**74.9**	79.2	**69.8**	**74.2**

Table 3: Results of our proposed method on Task2, causality localisation, compared with the baseline approaches on the MEDCAUS and FinCausal dataset.

for causality identification. We divide the dataset into train/test/validation sets based on the ratio 60:20:20.

We compare our model to the P-Wiki (Hidey and McKeown, 2016), which is a rule-based method, and bi-LSTM (Martínez-Cámara et al., 2017). Furthermore, since this task is closely related to the task of relation extraction, we compare our model to GCN, and C-GCN (Zhang et al., 2018), which use dependency tree information of the sentence.

The results are reported in Table 2. Our GCE-based classifier achieves the highest F1 and precision score on MEDCAUS, amongst all the models, followed closely by C-GCN. However, bi-LSTM shows the highest recall score. On FinCausal, our proposed model achieves the highest F1 and recall score, comparatively, while C-GCN hits the highest score on precision. Given the complexity and ambiguity of the projection of causal relation in natural language, taking both semantic and syntactic relations of a sentence improves the model. Hence, as suggested by the results, using both contextualised representation of a sentence and dependency relations of tokens of a sentence enriches the model, and results in obtaining more accurate prediction of causal relations.

4.4 Task2: Causality Localisation

This section covers the results of the performance of our proposed model, compared to other models, in terms of extracting cause and effect from textual data. MEDCAUS and FinCausal are used in this task for evaluation purposes. Each dataset are split into train/test/validation with the ratio of 60:20:20. For comparison, we report the results of the performance of each model in labelling each token with the proper tag. For this purpose, precision, recall, and F1 score are reported.

Similar to the Task1, we compare our model to bi-LSTM (Martínez-Cámara et al., 2017), GCN, and C-GCN (Zhang et al., 2018). Also, we compare our model to the proposed model of Lample et al. (2016), with two variations of using S-LSTM and ELMO (Peters et al., 2018) for contextual embedding.

The results of causality localisation are reported in Table 3. The experiments on MEDCAUS show that while bi-LSTM-CRF achieves better results in precision, it fails to gain high recall. On the hand C-GCN-CRF achieves highest recall, followed closely by our model. However, in F1 score, our model, outperforms the baselines. On FinCausal, bi-LSTM-CRF achieves the highest precision. However, our model achieves better recall and F1 score.

4.5 Results of ACE

In this section, we present the results of our ACE model on the task of adaptive causality identification and causality localisation. To this end, we

	MEDCAUS → BioCausal			MEDCAUS → SemEval			MEDCAUS → FinCausal		
Models	P	R	F1	P	R	F1	P	R	F1
bi-LSTM (Martínez-Cámara et al., 2017)	76.1	57.6	66.0	82.5	62.8	71.3	47.6	8.3	14.2
GCN (Zhang et al., 2018)	75.4	51.0	60.8	78.4	67.6	72.6	49.2	53.2	51.1
C-GCN (Zhang et al., 2018)	71.3	42.9	55.3	84.1	70.8	76.9	48.7	52.3	50.4
bi-LSTM+DA	75.6	58.9	66.2	81.6	69.5	75.1	47.9	61.1	53.7
GCN+DA	72.8	70.3	71.5	82.9	66.7	73.9	46.8	57.4	51.6
C-GCN+DA	78.4	55.5	65.1	81.9	71.0	76.1	**49.1**	54.6	52.7
CDAN (Long et al., 2018)	**85.5**	50.1	63.8	**84.6**	73.8	78.8	43.6	53.3	48.0
CDAN-E (Long et al., 2018)	**83.8**	55.0	66.4	81.2	74.2	77.6	48.3	63.3	54.8
ACE	74.3	**77.1**	**76.7**	**84.4**	**74.2**	**79.0**	47.4	**74.0**	**57.8**

Table 4: Results on our proposed adaptive causality encoder compared with the baselines for the task of causality identifier. The source dataset is MEDCAUS. BioCausal, SemEval, and FinCausal are considered as the target dataset.

	MEDCAUS → SemEval			MEDCAUS → FinCausal		
Models	P	R	F1	P	R	F1
bi-LSTM (Martínez-Cámara et al., 2017)	16.3	52.2	24.9	**64.1**	16.1	25.8
GCN (Zhang et al., 2018)	8.8	29.6	13.5	41.6	40.9	41.0
C-GCN (Zhang et al., 2018)	18.9	47.6	27.1	63.8	13.0	21.6
bi-LSTM+DA	**51.2**	42.0	46.2	44.9	39.4	41.9
GCN+DA	9.1	25.5	13.4	39.6	45.2	42.2
C-GCN+DA	45.1	42.1	43.6	40.0	**45.5**	42.6
CDAN (Long et al., 2018)	40.9	49.7	44.8	36.9	42.8	39.6
CDAN-E (Long et al., 2018)	47.3	40.6	43.7	36.8	42.6	39.5
ACE	42.3	**53.6**	**47.3**	42.2	43.2	**42.7**

Table 5: Results on our proposed adaptive causality encoder compared with the baselines for the task of causality localisation. The source dataset is MEDCAUS and target dataset are SemEval and FinCausal.

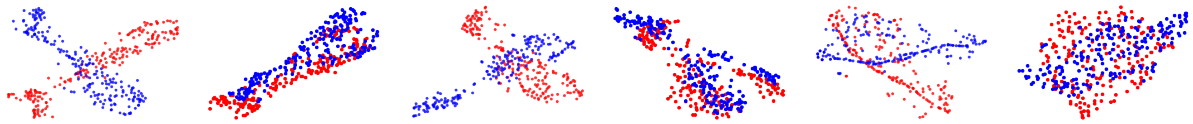
consider MEDCAUS as the source domain, and SemEval-10, BioCausal, and FINCAUSAL as target domains [7]. We compare our model to bi-LSTM (Martínez-Cámara et al., 2017), GCN and C-GCN (Zhang et al., 2018) as the baselines, their domain adaptive versions (indicated with "+DA" in the tables), and a state-of-the-art approach of conditional adversarial domain adaptation (CDAN and CDAN-E) (Long et al., 2018).

Table 4 summarises the results of our experiments of domain adaptive causality identification. As it can be seen, adding the domain adaptive strategy to the baselines improves their performance on all target datasets, by up to 39%. Furthermore, while CDAN achieves a better precision, it fails to balance the recall and performs poorly in terms of F1 score. On the other hand, our model (ACE; c.f. §3.2), outperforms all of the other models in recall and F1 score.

The results on applying domain adaptation method for the task of causality localisation is reported in Table 5. Applying our proposed domain adaptive model has improved the recall and F1 score of the baselines on both target datasets. While other models achieve better precision scores, our model consistently gains a better recall and F1 score, showing the superiority of our approach.

Visualisation Figure 2 visualises the effect of applying our proposed domain adaptation module (ACE; c.f. §3.2), to different target datasets. The extracted features ($f_{\theta_{enc}}$) of the source and target datasets are visualised using t-distributed Stochastic Neighbour Embedding (t-SNE) (Maaten and Hinton, 2008). The source and target datasets are shown in red and blue, respectively. In each subfigure, the features before and after applying ACE are represented on the left and right side, respectively. It is clear that, where the source and target domains data have different distributions, ACE

[7] Since BioCausal does not provide tags of cause and effect, this dataset was not used for domain adaptive causality localisation.



(a) MEDCAUS → SemEval (b) MEDCAUS → BioCausal (c) MEDCAUS → FinCausal

Figure 2: t-SNE visualisation of the domain adaptation task (with perplexity of 20 and Principal Decomposition Analysis (PCA) (Wold et al., 1987) initialisation). The source and target data are shown in red and blue, respectively. In each section, features before and after applying ACE are represented,on left and right side, respectively.

Causality Identification	MEDCAUS	1. Severe narrowings may cause chest pain (angina) or breathlessness during exercise or even at rest. 2. When the floor of the mouth is compressed, air is forced into the lungs.	✓ ×
	SemEval	1. Mechanical faults caused delays and cancellations on Wellington's suburban train services this morning 2. The overall damage caused by the destruction of land and property for the Wall's construction has taken many years to recover further.	× ✓
	FinCausal	1. Thomas Cook, one of many world's largest journey corporations, was based in 1841 to function temperance day journeys, and now has annual gross sales of 39 billion. 2. The judge's decision converted the arbitration award to a legal judgement and the sum, including interest accrued since 2013, soared to more than $9 billion.	× ✓
	BioCausal	1. For cost and convenience reasons other altered fractionation schedules have been adopted in routine practice. 2. The sequential technique also minimises the incidence of iris bleeding.	× ✓
Causality Localisation	MEDCAUS	1. **A high rate of consumption** can also lead to **cirrhosis, gastritis, gout, pancreatitis, hypertension, various forms of cancer, and numerous other illnesses.** 2. **The phlegm produced by catarrh** may either discharge or cause **a blockage** that may become chronic.	
	SemEval	1. He took a shower after using hair cream to avoid **skin irritation** from the **chemicals in the product**. 2. A **cigarette** set off a **smoke alarm**.	
	FinCausal	1. **The DGR in the Roth is lower at 5.4%** due primarily to its **holding of REITs**. 2. **Company tax receipts** were $4.6 billion higher than predicted, mainly due to **mining profits**, but Mr Frydenberg could not say how much was due to strong iron ore demand.	

Table 6: Examples of the performance of our proposed model in causality identification and localisation. The top section of the table provides examples for the first task. The ✓and × indicate identification of causal and no causal relation in the sentence, respectively. The bottom part of the table presents examples for the second task. We have used underline with red colour and dashed underline with blue colour to show cause and effect respectively.

matches the distributions, which greatly helps with improving the performance on the target data.

4.6 Qualitative Analysis

In this section, we demonstrate the capability of our proposed models in addressing the tasks of causality identification and localisation. To this end, for each task, two sentences from each dataset are presented in Table 6. The top section of the table provides examples for causality identification. The bottom section presents example for causality localisation. The examples suggests that our proposed models perform accurately on datasets with different distributional features.

5 Conclusion

In this work, we propose a new dataset for the task of causal identification and causal extraction from natural language text. We further propose a neural-based model for textual causality identification and localisation, which makes use of dependency trees. We then make use of adversarial training to adapt the causality identification and localisation models to new domains. Empirical results show that our method outperforms state-of-the-art models and their adapted versions.

References

Hanna Abi Akl, Dominique Mariko, and Estelle Labidurie. 2020. Semeval-2020 task 5: Detecting counterfactuals by disambiguation. *arXiv*.

Ning An, Yongbo Xiao, Jing Yuan, Jiaoyun Yang, and Gil Alterovitz. 2019. Extracting causal relations from the literature with word vector mapping. *Computers in biology and medicine*.

Eduardo Blanco, Nuria Castell, and Dan I Moldovan. 2008. Causal relation extraction. In *Lrec*.

Hong-Jie Dai, Po-Ting Lai, Yung-Chun Chang, and Richard Tzong-Han Tsai. 2015. Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization. *Journal of cheminformatics*.

Tirthankar Dasgupta, Rupsa Saha, Lipika Dey, and Abir Naskar. 2018. Automatic extraction of causal relations from text using linguistically informed deep neural networks. In *Annual SIGdial Meeting on Discourse and Dialogue*.

Tharini N De Silva, Xiao Zhibo, Zhao Rui, and Mao Kezhi. 2017. Causal relation identification using convolutional neural networks and knowledge based features. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*.

Quang Xuan Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *Conference on Empirical Methods in Natural Language Processing*.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*.

Clive WJ Granger. 1988. Some recent development in a concept of causality. *Journal of econometrics*.

Chikara Hashimoto. 2019. Weakly supervised multilingual causality extraction from wikipedia. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Oktie Hassanzadeh, Debarun Bhattacharjya, Mark Feblowitz, Kavitha Srinivas, Michael Perrone, Shirin Sohrabi, and Michael Katz. 2019. Answering binary causal questions through large-scale text mining: An evaluation using cause-effect pairs from human experts. *IJCAI19*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Workshop on Semantic Evaluations: Recent Achievements and Future Directions*.

Christopher Hidey and Kathy McKeown. 2016. Identifying causal relations using parallel wikipedia articles. In *Annual Meeting of the Association for Computational Linguistics*.

Dongyeop Kang, Varun Gangal, Ang Lu, Zheng Chen, and Eduard Hovy. 2017. Detecting and explaining causes from text for a time series event. *arXiv preprint arXiv:1707.08852*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Manolis Kyriakakis, Ion Androutsopoulos, Artur Saudabayev, et al. 2019. Transfer learning for causal sentence detection. *arXiv preprint arXiv:1906.07544*.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Shining Liang, Wanli Zuo, Zhenkun Shi, and Sen Wang. 2019. A multi-level neural network for implicit causality detection in web texts. *arXiv preprint arXiv:1908.07822*.

Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *Conference on Empirical Methods in Natural Language Processing*.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Annual meeting of the association for computational linguistics: system demonstrations*.

Eugenio Martínez-Cámara, Vered Shwartz, Iryna Gurevych, and Ido Dagan. 2017. Neural disambiguation of causal lexical markers based on context. In *International Conference on Computational Semantics Short papers (IWCS)*.

Paramita Mirza and Sara Tonelli. 2016. Catena: Causal and temporal relation extraction from natural language texts. In *International Conference on Computational Linguistics: Technical Papers*.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. Joint reasoning for temporal and causal relations. In *Annual Meeting of the Association for Computational Linguistics*.

Rajaswa Patil and Veeky Baths. 2020. Cnrl at semeval-2020 task 5: Modelling causal reasoning in language with multi-head self-attention weights based counterfactual detection. *arXiv preprint arXiv:2006.00609*.

Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Conference on empirical methods in natural language processing (EMNLP)*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*.

Mehwish Riaz and Roxana Girju. 2013. Toward a better understanding of causality between verbal events: Extraction and analysis of the causal power of verb-verb associations. In *SIGDIAL 2013 Conference*.

Mehwish Riaz and Roxana Girju. 2014. Recognizing causality in verb-noun pairs via noun and verb semantics. In *EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Mateo Rojas-Carulla, Marco Baroni, and David Lopez-Paz. 2017. Causal discovery using proxy variables. *arXiv preprint arXiv:1702.07306*.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*.

Xiaoyu Yang, Stephen Obadinma, Huasha Zhao, Qiong Zhang, Stan Matwin, and Xiaodan Zhu. 2020. Semeval-2020 task 5: Counterfactual recognition. In *International Workshop on Semantic Evaluation (SemEval-2020), Barcelona, Spain*.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Conference on Empirical Methods in Natural Language Processing*.

Sendong Zhao, Meng Jiang, Ming Liu, Bing Qin, and Ting Liu. 2018. Causaltriad: Toward pseudo causal relation discovery and hypotheses generation from medical text data. In *ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*.

Sendong Zhao, Quan Wang, Sean Massung, Bing Qin, Ting Liu, Bin Wang, and ChengXiang Zhai. 2017. Constructing and embedding abstract event causality networks from text snippets. In *ACM International Conference on Web Search and Data Mining*.

Automated Detection of Cyberbullying Against Women and Immigrants and Cross-domain Adaptability

Thushari Atapattu[1], **Mahen Herath**[2], **Georgia Zhang**[1] and **Katrina Falkner**[1]
[1]School of Computer Science, The University of Adelaide, Adelaide, Australia
[2]Department of Computer Science & Engineering, University of Moratuwa, Katubedda, Sri Lanka
email: thushari.atapattu@adelaide.edu.au

Abstract

Cyberbullying is a prevalent and growing social problem due to the surge of social media technology usage. Minorities, women, and adolescents are among the common victims of cyberbullying. Despite the advancement of NLP technologies, the automated cyberbullying detection remains challenging. This paper focuses on advancing the technology using state-of-the-art NLP techniques. We use a Twitter dataset from SemEval 2019 - Task 5 (HatEval) on hate speech against *women* and *immigrants*. Our best performing ensemble model based on DistilBERT has achieved 0.73 and 0.74 of F1 score in the task of classifying hate speech (Task A) and aggressiveness and target (Task B) respectively. We adapt the ensemble model developed for Task A to classify offensive language in external datasets and achieved ∼0.7 of F1 score using three benchmark datasets, enabling promising results for cross-domain adaptability. We conduct a qualitative analysis of misclassified tweets to provide insightful recommendations for future cyberbullying research.

1 Introduction and Motivation

Cyberbullying is *"the repetitive use of aggressive language amongst peers with the intention to harm others through digital media"* (Rosa et al., 2019). Due to the surge of social media technology use, cyberbullying has become a prevalent and growing social problem. Unlike in the physical environment, cyberspace, in particular, online social platforms are not yet evolved sufficiently to prevent people from communicating without disclosing identities, spreading rumours, and harassing others. The risk of and potential consequences caused by cyberbullying are critical including both physical and mental health risk to victims. The impact and consequences are common to all generations (e.g. young, elderly) including emotional and psychological distress, decline in personal/academic development, anti-social behaviour, and, potentially, suicide.

The criticality of this societal problem is demonstrated from a study by Yale University, commenting *"cyberbullying victims are 2 to 9 times more likely to consider committing suicide"* across the globe.[1] Within Australia, the eSafety Commissioner comments *"one in every five Australian children aged eight to seventeen are victims of cyberbullying (2018)"*.[2] Adolescents, minorities (e.g. refugees, LGBTQI) and women are among common targets of cyberbullying. According to Bullying Statistics[3], over half of adolescents are victims of cyberbullying and about the same percentage are involved in bullying.

Despite recent research advancement in *hate speech* detection (Fortuna and Nunes, 2018), automated identification of *cyberbullying* attempts (i.e. repetitive hate speech against an individual or a group) remains as a challenging subtask of NLP. Due to diverse variants of language (e.g. hate, intimidation, sarcasm, metaphors) used by bullies and the evolution of language (e.g. slang), particularly among adolescents, the automated detection of cyberbullying is extremely challenging. The example below appears to be misogynistic as it includes the term *'b***h'*; however, it is manually classified as not misogyny since the slang *'gay a*s b***h'* is commonly used for a male or gay person.

> "*you a **gay a*s b***h** who seeks attention, STOP! I knew ever since you gonna switch up on me... I guess you did F***ING SNAKE A*S H*E!"*

[1]https://theorganicagency.com/blog/life-death-consequences-cyber-bullying/

[2]https://www.theguardian.com/society/2018/oct/03/one-in-five-australian-children-are-victims-ofcyberbullying-e-safety-commissioner-says

[3]http://www.bullyingstatistics.org/content/cyber-bullying-statistics.html

To mitigate the research and social problem of cyberbullying, this paper focuses on advancing the technology to classify cyberbullying using state-of-the-art NLP techniques. As a case study, we focus on cyberbulling against women and immigrants. Accordingly, our first research question (RQ1) asks, *Can we build machine learning models to outperform current cyberbullying classification systems on women and immigrants?*. The findings of RQ1 will lead us to explore the limitations of our models and explanations for misclassification. Hence, our second research question (RQ2) investigates, *What is the content of misclassified tweets and how can we categorise them?*. Finally, to evaluate the validity of our models across external cyberbullying/hate speech datasets, our third research question (RQ3) investigates, *Can we successfully validate machine learning models developed for cyberbullying detection within the context of women and immigrants for other benchmark datasets?*.

To answer our research questions, we utilise a Twitter dataset developed for SemEval 2019 - Task 5 (HatEval) (Basile et al., 2019) that includes labels for three sub tasks: 1) hate speech, 2) aggressiveness, and 3) target (individual or group). We adopt a mixed-method study, using a combination of the building of machine learning models (RQ1 & RQ3) and qualitative content analysis (RQ2) as our methodology. We make the following main contributions:

- We developed and evaluated cyberbullying classification models using state-of-the-art NLP technology. Even though our model performance on Task A is either equal or slightly lower than baselines, we outperformed all previous best systems and baselines on Task B. Therefore, our ensemble models based on DistilBERT (Sanh et al., 2019) serves as the best system as yet to classify aggressiveness and target (Task B).

- We conducted a qualitative study to categorise misclassified tweets into meaningful codes. We distinguished six categories: lack of context (CNTX), gender-related issues (GEND), issue with resolving slangs (SLNG), issues in the original annotation (ERROR), misclassified by our model (MSCL), and issues not belong to any category (OTHER) emerged from our data, establishing a point of reference for future researchers in cyberbullying, particularly, within the context of minorities (e.g. women, LGBTQI, immigrants).

- We adopted our best pre-trained model to evaluate other benchmark datasets, including OffensEval challenge (Zampieri et al., 2019, 2020) and Hate Offense task (Davidson et al., 2017). Our model generalised reasonably well ($\sim$0.7) with both tasks, contributing to developing a generalised model across different cyberbullying-related tasks.

2 Background and Related Work

Cyberbullying is a complex phenomenon that needs multiple psychological, linguistic, and social theories to understand its nature. The identification of cyberbullying is inherently more complex even for humans (except victims) as it involves repetitive behaviour, peer-oriented nature, and intentionality to harm. Therefore, we utilise a definition stated in a recent systematic literature review on cyberbullying (Rosa et al., 2019) as *"repetitive use of aggressive language amongst peers with the intention to harm others through digital media"*.

Some recent studies (Fortuna and Nunes, 2018) including WOAH[4] (previously known as ALW) workshop (Roberts et al., 2019) have focused on *hate speech* detection as a more general field. Despite recent advancement in *hate speech detection*, recognising cyberbullying in everyday problems is primarily manual based on victim reports or manual moderation. Recent studies rely on contextual features such as demography, social network, and sentiments/emotions as features to train cyberbullying classifiers (Chatzakou et al., 2019).

Conversely, some related workshops such as TRAC (Kumar et al., 2018) and challenges such as HatEval (Basile et al., 2019), OffensEval (Zampieri et al., 2019, 2020) contributed to advance the research field by developing systems using cutting-edge NLP techniques like Universal Encoder - Fermi (Indurthi et al., 2019), LT3 (Bauwelinck et al., 2019), ensemble of deep learning models like OpenAI's GPT and Transformer models (Team NLPR@SAPOL (Seganti et al., 2019)), and BERT (NULI (Liu et al., 2019)). Some of these systems have surpassed baselines and earned recognition as the best-performing systems in specific subtasks (e.g. NULI achieved 0.82 of F1 score and ranked 1st place in subtask A to classify offensive language

[4]https://www.workshopononlineabuse.com/home

while it ranked only in 18th place for subtask C to classify targets such as individuals, group).

Despite the promise of current systems, these models are not consistent enough to perform reasonably well within all sub tasks of cyberbullying (i.e. hate speech, aggressiveness and target). Additionally, these models were not validated across other cyberbullying-related tasks to ensure generalisability. Related literature also lacks comprehensive contributions to draw implications on why machine learning models fail to improve further. Our work focuses on addressing these three drawbacks.

3 Research Methodology

Research Questions. Our research is guided by three research questions,

- **RQ1:** Can we build machine learning models to outperform current cyberbullying classification systems?
- **RQ2:** What is the content of misclassified tweets, and how can we categorise them?
- **RQ3:** Can we successfully validate machine learning models developed for cyberbullying detection within the context of women and immigrants for other benchmark datasets?

Dataset. We utilise a dataset collected from Twitter during July to September 2018 for SemEval 2019 - Task 5 (HatEval) challenge (Basile et al., 2019). This challenge was organised to advance the technology to classify cyberbullying against women and immigrants. Tweets were collected both from English and Spanish language. We utilise only the English dataset in this paper. The dataset contains a set of tweets and their labels; HS - Hate Speech (0 - No, 1 - Yes), TR - Target Range (0 - generic group, 1 - individual), AG - Aggressiveness (0 - No, 1 - Yes). The challenge was divided into two subtasks, Task A - classification of HS, and Task B - classification of AG and TR. The dataset was labelled via AllCloud crowdsourcing platform and added two more experienced annotators to determine the final labels. Inter-rater reliability for HS, TR, and AG is 0.83, 0.7, 0.73 respectively. The dataset consists of a total of 13,000 tweets with 10,000 for training set (5,000 each for women and immigrant) and 3,000 for test set (1,500 each for women and immigrant). Table 1 of the work by Basile et al. (2019) demonstrates more information about data distribution.

3.1 Methods

We adopt a mixed-method study, using a combination of the building of machine learning models (RQ1 RQ3) and content analysis (RQ2) as our methodology.

Pre-processing. We conducted text preprocessing using standard techniques including tokenisation and removal of non-ASCII characters such as decoding emoticons[5]. Additionally, other preprocessing steps such as removal of punctuations and shortened URLs were performed while fine-tuning deep learning based models like DistilBERT (Sanh et al., 2019). We retained hashtags as these were important features of our models.

Building of machine learning models. We adopted state-of-the-art NLP and deep learning techniques for text classification to solve cyberbullying detection problem, and built our models using DistilBERT (Sanh et al., 2019), a lighter and a faster pre-trained language model based on BERT (Devlin et al., 2018). To answer RQ1 through model comparisons, we utilised MFC and top-ranked systems in each task of HatEval challenge (Basile et al., 2019) as our baselines.To answer RQ3, we apply our pre-trained models on HatEval into other benchmark datasets related to cyberbullying. For this, we utilise three external datasets developed for SemEval Task 12 - OffensEval2020 (Zampieri et al., 2020), SemEval Task 6 - OffensEval2019 (Zampieri et al., 2019) and Hate Offensive language detection by Davidson et al. (2017).

Content analysis. We adopted open coding (Corbin and Strauss, 1990), a qualitative content analysis technique as our method to answer RQ2 on exploring the content of misclassified tweets and categorisation them into a coding schema.

4 Model Description

4.1 Ensemble model - Task A

To address the Task A we created three classification models named A, B and C (see Figure 1) based on the DistilBERT model with a sequence classification head on top (Sanh et al., 2019). An imbalanced subset of training data where the majority class was positive was used to train model

[5]https://github.com/carpedm20/emoji

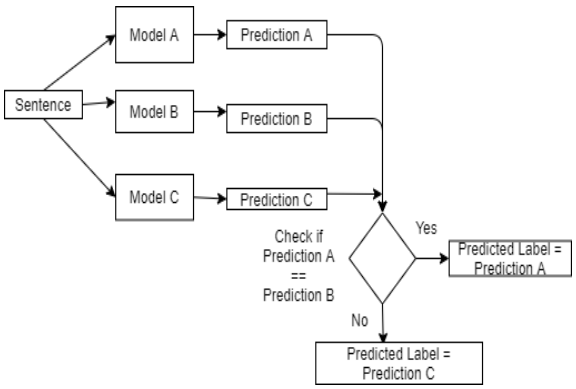


Figure 1: Deriving the final labels of Task A

A, and an imbalanced subset of training data where the majority class was negative was used to train model B. Inspired by the approach described in Khoussainov et al. (2005), model C was trained on a balanced subset of training data which were classified differently by the biased classifiers A and B. We fine tuned all three classifiers with a learning rate of 5e-05 for 3 epochs using a batch size of 32.Finally, we used simple voting to create an ensemble classifier combining the models A, B and C.

4.2 Ensemble model - Task B

Task B can be modelled as a multi-class (i.e. 5 classes) classification problem with the individual classes being (HS=0,TR=0,AG=0), (HS=1, TR=1, AG=0), (HS=1, TR=1, AG=1), (HS=1, TR=0, AG=1), and (HS=1, TR=0, AG=0) (Gertner et al., 2019). We developed 5 binary classifiers, one for each class, using the DistilBERT model with a sequence classification head on top. Each classifier was fine tuned with a learning rate of 5e-5 and a batch size of 32 for 3 epochs. We then combined the predictions from these classifiers using probabilities to derive the final class labels.

If only one classifier predicted a given data instance as positive, we assigned the class label of that classifier to the data instance. Whenever several classifiers predicted the positive class label for a given instance, we selected the prediction with the highest probability. If all the classifiers predicted the negative class label for a given instance, we selected the prediction with the lowest probability.

5 Results and Discussion

Evaluation Metric. To calculate the classification effectiveness, we use different metrics in each subtask. Task A uses the *macro-averaged F1 score* while Task B uses *Exact Match Ratio (EMR)* along with *macro-averaged F1 score* (Basile et al., 2019).

- **F1 Score.** The harmonic mean of precision and recall where precision is the proportion of predicted positive instances that are actually positive while recall is the proportion of actual positive instances that are predicted as positive.

- **Exact Matching Ratio (EMR).** Since Task B is a multi-label classification problem, EMR is calculated by combining all the dimensions (i.e. HS, TR, AG) to be predicted. The calculation of EMR is discussed in Basile et al. (2019).

Baselines. To evaluate our models (see Section 5.1), we utilise top-ranked HatEval systems and a system used by Basile et al. (2019) as our baselines,

1. **Task A**. Fermi (Indurthi et al., 2019) using the SVM model with Google's Universal Sentence Encoder (Cer et al., 2018) (refer as 'SVM+USE') surpassed SVC and MFC baselines of HatEval challenge.

2. **Task B**. LT3 (Bauwelinck et al., 2019) ranked top in Task B.

3. **MFC baseline**. MFC is a trivial model that assigns the most frequent label in training set to all instances in the test set.

5.1 Answering RQ1 - Model Evaluation

The performance of our ensemble model using the official HatEval test set is shown in Figure 2. The results demonstrate that our ensemble model has achieved 0.49 F1 score for Task A. In Task A, even though we outperformed MFC baseline (F1 score = 0.37), our scores did not exceed the best HatEval system - Fermi (F1 score = 0.65) (Indurthi et al., 2019). Nevertheless, our Task A performance scores are not promising for real-world adoption.

Conversely, our ensemble model has obtained 0.62 of F1 score for task B which exceeds the best systems of HatEval Task B - LT3 (Bauwelinck et al., 2019) (F1 score = 0.47) and MFC baseline (F1 score = 0.42) (Basile et al., 2019). In Task B of HatEval, no system has been able to outperform the EMR score of MFC baseline, which achieved 0.58 of EMR (Note: *Exact Matching Ratio* was the metric used for HatEval Task B evaluation). LT3 system and our ensemble model both equally achieved

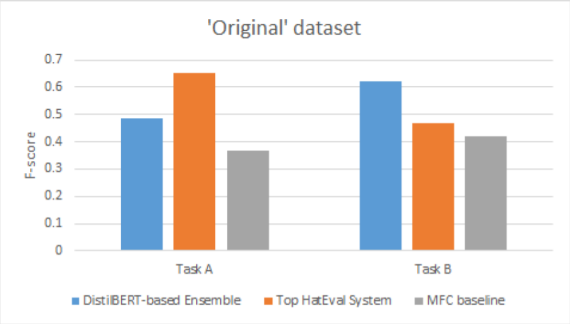


Figure 2: Model performance of Task A & B using 'original' HatEval test dataset

0.57 of EMR which ranked us in the top place for Task B followed by MFC baseline. Since our DistilBERT-based ensemble model achieved an F1 score over 0.9 in another cyberbullying-related task (SemEval Task 12 - OffenseEval 2020) (Zampieri et al., 2020)(Herath et al., 2020), we further analysed the peculiar behaviour of model performance with HatEval challenge by unpacking the dataset.

We plotted the percentages of tweets annotated as having hate speech when some common hashtags or derogatory tokens (e.g. #buildthatwall, b***h) were found in tweets. Figure 3a) depicts the variation of data across training, dev and test sets. According to Figure 3a), it appears that training and dev set are slightly similar yet drastically different from the test set. For example, it appears that the likelihood of tweets with the token '#buildthatwall (token 1)' being annotated as having hate speech is 100% in train and dev set, however, it is approximately 20% in the test set.

In order to examine whether discrepancies in the dataset had any impact on the poor performance, we merged development, training and test sets, shuffled the rows, and randomly split them again (referred to as 'adjusted' dataset) according to the proportions in the 'original' HatEval dataset (see Section 3 - 'dataset'). Figure 3b) demonstrates that there was a disparity with data distribution in the 'original' dataset. For example, in the 'adjusted' dataset, the percentage of '#buildthatwall' being annotated as having hate speech is approximate (∼60%) across train, dev and test sets. This finding led us to train our models with 'adjusted' dataset and fine-tuned the parameters. Figure 3b) and Figure 4 depicts the new data distribution and model performance using 'adjusted' dataset respectively.

According to Figure 4, our ensemble models have achieved 0.73 of F1 score for Task A and 0.75 of F1 score for Task B on 'adjusted' test set. We also achieved 0.62 EMR for Task B on test set. Due to the difficulty in replicating LT3 system (Bauwelinck et al., 2019) to train on 'adjusted' dataset, we obtained performance of 'SVM+USE' model (Indurthi et al., 2019) using our 'adjusted' dataset. As shown in Figure 4, our model and baseline demonstrated equal performance in Task A. Conversely, our model outperforms 'SVM+USE' baseline by a margin of 0.06 in Task B. As mentioned in Section 4, we used 3 epochs, a batch size of 32 and a learning rate of 5e-5 to train our models.

RQ1: We can automatically classify cyberbullying against women and immigrant with an F1 score of 0.73 and 0.75 in Task A (hate speech) and Task B (aggressive and targeted) respectively.

The primary focus of our research is on improving the *recall*, i.e. to correctly identify tweets that are cyberbullying attempts against women and immigrants as it will eventually contribute to safe cyberspace for minorities. We have achieved 0.73 and 0.76 of *recall* for Task A and B respectively using 'adjusted' dataset compared to low recall of baseline systems. We are also interested in controlling true negatives, i.e. tweets that are not actually cyberbullying but are identified as positive. We exceed precision of 0.73 in both tasks using our DistilBERT-based ensemble models. Otherwise, incorrect classification of cyberbullying will have an impact on the reputation of social media platforms, particularly for freedom of speech.

5.2 Answering RQ2 - Content Analysis of Misclassified Tweets

To answer our RQ2, we extracted misclassified tweets (task A & B) from our ensemble model. A content analysis method ('open coding') (Corbin and Strauss, 1990) has been adopted. The second author manually categorised 10 random misclassified tweets into three meaningful codes: gender-related issues (GEND), context-related issues (CNTX), and slangs (SLNG). After defining initial codes, two annotators (first and third author who are experienced in cyberbullying context) trialed them on a random sample of 299 misclassified tweets (population is 626 tweets), resulting in a confidence interval of 4.1 at a confidence level of 95%. To measure the inter-annotator agreement we used the Kappa statistic.Due to the complex nature of

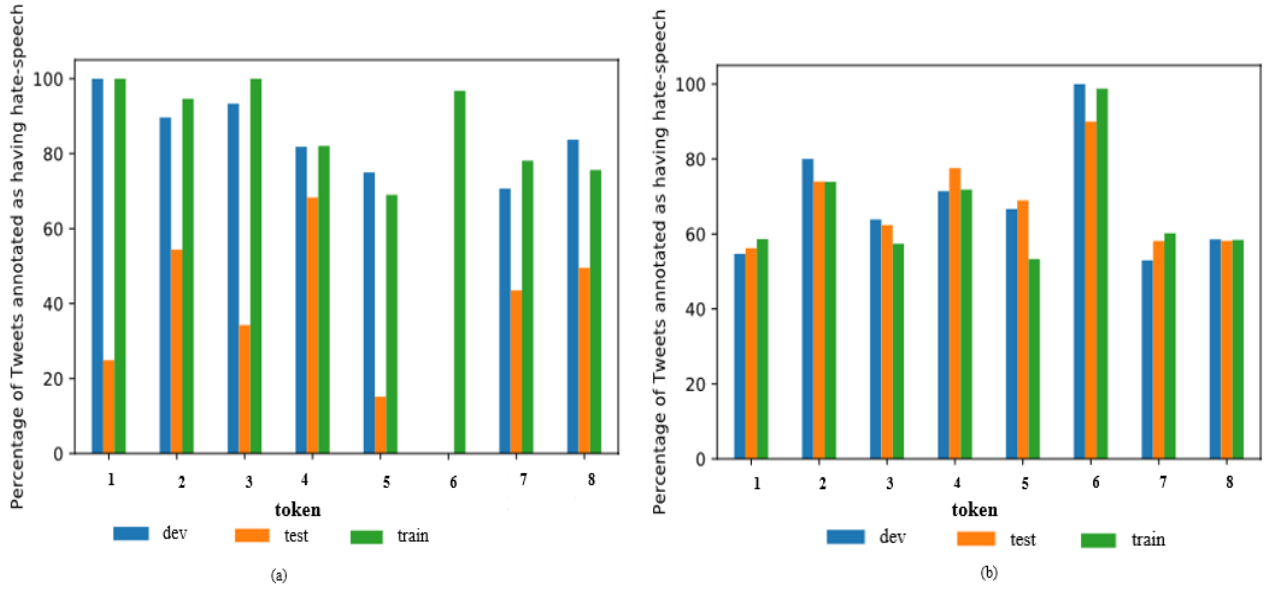


Figure 3: Variation of data across training, dev and test sets in (a) 'original' (b) 'adjusted' dataset; 1:#buildthatwall, 2:#buildthewall, 3:#nodaca, 4:#sendthemback, 5:#stoptheinvasion, 6:#womens**k, 7:b***h, 8:h*e

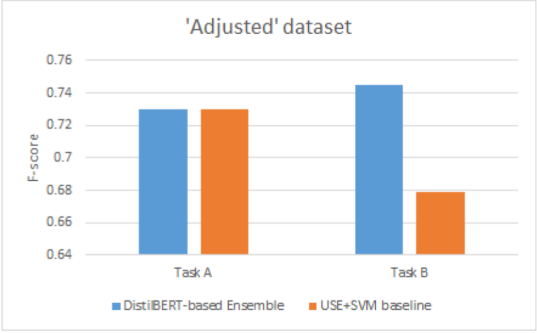


Figure 4: Model performance of Task A & B using 'adjusted' HatEval test dataset

cyberbullying phenomenon and availability of multiple codes to annotate, we failed to reach a reasonable inter-rater agreement. To overcome this, we refined our codes until we reach an agreement on a coding scheme that contained codes for all misclassified tweets in our sample. Finally, we added three additional codes: errors in original annotation (ERROR), misclassified by our model (MSCL), and not belong to any category (OTHER) when both annotators agree that original (HatEval) annotation is dubious, predicted label is incorrect, and when all other possibilities have been exhausted respectively. Table 1 shows the finalised set of codes along with their frequency distribution (%).

Our results demonstrate that the lack of contextual information to resolve pronouns or user names in tweets to determine 'gender' (i.e. whether the target is *women*) is one of the common reasons for misclassification. Based on the frequency distribution ('last column' in Table 1), the most frequent category of misclassification is 'CNTX'. Lack of contextual information is a widely raised constraint within the majority of previous works which aligns with our findings. The least frequent category of misclassification is 'SLNG'. One possible explanation for this behaviour could be due to the dataset is extracted from an 'adult' group, and they are less likely to introduce new slang words compared to adolescents. Also, our results suggest that 3% of misclassified tweets are due to 'errors' (∼10 tweets) in the original annotations.

Conversely, we admit that our model predicted inaccurate labels in 3% of cases (∼10 tweets). Our findings suggest that 30% of instances belong to 'OTHER' category. Through manual inspection, we observed that this might be due to reasons like sarcasm, swearing with friends, abbreviations, complaints, and negations. However, the analysis reported in this paper is not comprehensive to include adequate evidence to report subcategories.

RQ2: Misclassified tweets can be categorised into six types, with the context-related issues ('CNTX') being the most frequent reason for misclassification, followed by issues to resolve gender ('GEND') and slang ('SLNG').

Code	Definition	Example	Explanation	(%)
GEND	Gender-related issues	*You seem like a h*e Ok b***h? Did I ever deny that? Nope, Next.*	Misogynistic if 'you' refers to a female	11
CNTX	Lack of context	*@user you deserve allll the hate you get you h*e a*s b***h, out here being a damn Hypocrite you and cash some damn FAKES. H**s*	Misogynistic if '@user' refers to a female	44
SLNG	Issues in resolving slang	you a gay a*s b***h who seeks attention, STOP! I knew ever since you gonna switch up on me... I guess you did F***ING SNAKE A*S H*E!	Non-misogynistic if 'gay a*s b***h' slang is resolved	9
ERROR	Issues in original annotation	@user It means <religion> will show them how to rape/abuse women 24/7!	Targeted to immigrants	3
MSCL	Misclassified by our model	Europe is being invaded by third world "refugees" Continue to Pray for them	Targeted refugees	3
OTHER	Not belong to any category	REFUGEES NOT WELCOME	Targeted refugees if negation is recognised	30

Table 1: Coding reference of misclassified tweets.

5.3 Answering RQ3 - Cross-task Evaluation

To answer our RQ3 about the generalisability of our models on different cyberbullying-related tasks, we applied and tested our pre-trained ensemble model in other three tasks, 1) SemEval 2020 - Task 12 (OffensEval2020) (Zampieri et al., 2020), 2) SemEval 2019 - Task 6 (OffensEval2019) (Zampieri et al., 2019), and 3) Hate & Offensive language (refer as 'Hate & Offense') dataset (Davidson et al., 2017).

1. **OffensEval datasets (Zampieri et al., 2019, 2020)**. These datasets include three subtasks to determine whether a tweet expresses cyberbullying based on whether it is, 1) offensive or not, 2) targeted or not, and 3) if targeted, whether it is toward an individual, group, or other.

2. **Hate & Offense dataset (Davidson et al., 2017).** This dataset also has three subtasks to determine whether a tweet include hate and offensive language based on, 1) hate speech or not, 2) offensive but not hate speech, and 3) neither offensive nor hate speech.

The tasks of these two datasets were different from HatEval challenge except the first subtask to determine hate (or offensive) language. Therefore, we report the results of cross-domain validation using Task A (i.e. hate speech or not) only.We extracted a random sample of 2,971 tweets from each dataset to align with the test size of our original Task A when the official test set was unavailable publicly. 2 shows the outcome using our pre-trained ensemble model (Task A).

Dataset	Sample size	Acc.	P	R	F1
OffensEval2020	3887	0.74	0.72	0.74	0.68
OffensEval2019	860	0.68	0.66	0.68	0.67
Hate & Offense	2971	0.70	0.74	0.70	0.69

Table 2: Performance (weighted average) of our pre-trained Task A model on other cyberbullying-related tasks; Acc.:Accuracy, P:Precision, R:Recall.

Current state of the art models have reportedly achieved F1 scores of 0.82, 0.92 and 0.90 for OffensEval2019, OffensEval2020 and Hate & Offense datasets respectively (Zampieri et al., 2019, 2020)(Davidson et al., 2017). According to Table 2, we have achieved a satisfactory performance with approximately 0.7 of accuracy/F1 score for all task pairs (i.e. training on HatEval dataset and testing on another dataset).These results suggest that our pre-trained ensemble model on HatEval is generalised reasonably well (Accuracy/F1 score $\sim$0.7) when classifying *hate speech* irrespective of the context (e.g. misogyny etc.). Due to the misalignment between datasets, we did not apply our models to other tasks of external datasets.

RQ3: Our pre-trained models from HatEval dataset can automatically classify hate speech in other benchmarking datasets with a reasonable accuracy (∼0.7).

6 Discussion

The ultimate goal of our work is to advance the technology to detect and classify cyberbullying using state-of-the-art NLP techniques, with the long-term aim of enabling social media as a safe space for all users. We developed DistilBERT-based ensemble model per task as a basis to answer our RQ1. With an initial poor performance using a test set of 'original' HatEval dataset, we suggest developing a novel version of the original dataset (i.e. 'adjusted') through merging, shuffling and splitting. The *'adjusted'* dataset contributed to better performance of F1 score of 0.73 and 0.74 for Task A and B respectively.

The six categories of misclassified tweets that emerged from our qualitative analysis (RQ2) build a point of reference for the content of such misclassifications. This initial categories can help researchers to understand the grounds to improve automated cyberbullying classification. Also, the categories identified through this research can serve as a guide which could extend as a conceptual framework for future qualitative and quantitative cyberbullying research. Additionally, the categories along with the frequencies that we report in this work provide implications for researchers to collect, annotate, and revise their datasets that could minimise the likelihood of misclassification produced by machine learning models including providing additional contextual information about data. Conversely, this raises new research questions on whether we could improve the performance of machine learning models further without relying on demographic data such as *gender* and data on language evolution such as *out-of-vocabulary slang* and *abbreviations*.

The findings from our RQ3 on generalisability of pre-trained models on other cyberbullying-related tasks demonstrated reasonable accuracy (∼0.7). A possible explanation of not achieving more could be due to pre-trained models might biased within women and immigrant context (e.g. specific hashtags, misogyny) and not be the best option for classifying 'general' offense-related tasks. As a solution, future models could augment data from general as well as specific datasets (e.g., racial (Davidson et al., 2019), gendered (Kumar et al., 2020)).

In addition to the lack of contextual information that limits our model improvement further, this research is subject to implicit bias of annotators when judging categories to answer RQ2. As a solution, our future work will incorporate a semi-automated approach for misclassification annotation by reusing readily available lexical resources like MRC psycholinguistic database (Coltheart and Wilson, 1987), LIWC (Pennebaker et al., 2001) to obtain initial codes and employ at least three annotators to refine the codes. Furthermore, our 'adjusted' dataset may not provide a robust solution in terms of replicability. Therefore, we intend to create a couple of 'adjusted' datasets and report the average of performance in our future works. We also share our current 'adjusted' dataset to enable replication of experiments.

In summary, we propose that future cyberbullying classification models need to concentrate on incorporating state-of-the-art solution to common NLP problems like language evolution, sarcasm detection, and pronoun resolution. Additionally, future research should also focus on advancing the prediction of demographic information such as gender, age, and personality from data within an ethical framework without reidentifying Twitter profiles.

7 Conclusions

Due to massive participation in social media, manual moderation of cyberbullying is an extremely labour-intensive task which leads to delay in taking action against bullies while protecting victims. Accordingly, automated classification of cyberbullying emerged and remains as a challenging NLP task. This research contributes to develop machine learning models for cyberbullying classification. Through a qualitative content analysis, we also contributed to develop a coding schema to deepen the understanding of misclassifications produced by models, enabling future researchers to minimise the impact of data for poor model performance. When social media platforms are equipped with effective cyberbullying detection models, victimised communities will be able to discuss their concerns openly, without harassment.

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel

Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63.

Nina Bauwelinck, Gilles Jacobs, Véronique Hoste, and Els Lefever. 2019. LT3 at SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter (hatEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 436–440. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder.

Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, Athena Vakali, and Nicolas Kourtellis. 2019. Detecting cyberbullying and cyberaggression in social media. *ACM Trans. Web*, 13(3).

1939 Coltheart, M. (Max) and 1939 Wilson, Michael John. 1987. MRC psycholinguistic database machine usable dictionary : expanded shorter oxford english dictionary entries / max coltheart and michael wilson. Oxford Text Archive.

J. Corbin and A. Strauss. 1990. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology*, 13:3–21.

T. Davidson, Dana Warmsley, M. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *ICWSM*.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

Abigail S. Gertner, John C. Henderson, Amy Marsh, Elizabeth M. Merkhofer, Ben Wellner, and Guido Zarrella. 2019. Mitre at semeval2019 task 5: Transfer learning for multilingual hate speech detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, page 453– 459.

Mahen Herath, Thushari Atapattu, Hoang Dung, Christoph Treude, and Katrina Falkner. 2020. AdelaideCyC at SemEval-2020 Task 12: Ensemble of Classifiers for Offensive Language Detection in Social Media. In *Proceedings of SemEval*.

Vijayasaradhi Indurthi, Bakhtiyar Syed, Manish Shrivastava, Nikhil Chakravartula, Manish Gupta, and Vasudeva Varma. 2019. FERMI at SemEval-2019 task 5: Using sentence embeddings to identify hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 70–74.

Rinat Khoussainov, Andreas Heß, and Nicholas Kushmerick. 2005. Ensembles of biased classifiers. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 425–432, New York, NY, USA. Association for Computing Machinery.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2020. Evaluating aggression identification in social media. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 1–5. European Language Resources Association (ELRA).

Ritesh Kumar, Atul Kr. Ojha, Marcos Zampieri, and Shervin Malmasi, editors. 2018. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Association for Computational Linguistics.

Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91. Association for Computational Linguistics.

J. Pennebaker, L. Francis, and R. Booth. 2001. Liwc: Linguistic inquiry and word count.

Sarah T. Roberts, Joel Tetreault, Vinodkumar Prabhakaran, and Zeerak Waseem, editors. 2019. *Proceedings of the Third Workshop on Abusive Language Online*. Association for Computational Linguistics.

H. Rosa, N. Pereira, R. Ribeiro, P.C. Ferreira, J.P. Carvalho, S. Oliveira, L. Coheur, P. Paulino, A.M. Veiga Simão, and I. Trancoso. 2019. Automatic cyberbullying detection: A systematic review. *Computers in Human Behavior*, 93:333 – 345.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Alessandro Seganti, Helena Sobol, Iryna Orlova, Hannam Kim, Jakub Staniszewski, Tymoteusz Krumholc, and Krystian Koziel. 2019. Nlpr@srpol at semeval-2019 task 6 and task 5: Linguistically enhanced deep learning offensive sentence classifier. *CoRR*, abs/1904.05152.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020).

The Influence of Background Data Size on the Performance of a Score-Based Likelihood Ratio System: A Case of Forensic Text Comparison

Shunichi Ishihara
Speech and Language Laboratory
The Australian National University
shunichi.ishihara@anu.edu.au

Abstract

This study investigates the robustness and stability of a likelihood ratio–based (LR-based) forensic text comparison (FTC) system against the size of background population data. Focus is centred on a score-based approach for estimating authorship LRs. Each document is represented with a bag-of-words model, and the Cosine distance is used as the score-generating function. A set of population data that differed in the number of scores was synthesised 20 times using the Monte-Carol simulation technique. The FTC system's performance with different population sizes was evaluated by a gradient metric of the log–LR cost (C_{llr}). The experimental results revealed two outcomes: 1) that the score-based approach is rather robust against a small population size—in that, with the scores obtained from the 40~60 authors in the database, the stability and the performance of the system become fairly comparable to the system with a maximum number of authors (720); and 2) that poor performance in terms of C_{llr}, which occurred because of limited background population data, is largely due to poor calibration. The results also indicated that the score-based approach is more robust against data scarcity than the feature-based approach; however, this finding obliges further study.

1 Introduction: The Likelihood Ratio Framework and Forensic Text Comparison

The likelihood ratio (LR) conceptual framework has been studied for its effect on various types of forensic evidence; it was mathematically shown that, with some very reasonable assumptions, the LR is the only way of assessing the uncertainty inherited in evidential evaluation (Aitken, 2018; Aitken and Taroni, 2004; Good, 1991). It is becoming recognised as the logical and legally correct framework for both analysing forensic evidence and presenting it in court (Balding, 2005; Evett et al., 1998; Marquis et al., 2011; Morrison, 2009; Neumann et al., 2007). Yet, some argue that the LR is one possible tool for communicating to decision makers (Lund and Iyer, 2017: 1). Although forensic text comparison (FTC) currently lags behind other forensic sciences, some studies have demonstrated that linguistic text evidence can be properly analysed using the LR framework (Ishihara, 2014, 2017a, 2017b).

In the LR framework, instead of assessing the probabilities of two competing hypotheses given the evidence, the probabilities of observing the evidence (E) are assessed given the hypotheses: the prosecution hypothesis (H_p) against the defence hypothesis (H_d) (Aitken and Stoney, 1991; Aitken and Taroni, 2004; Robertson et al., 2016). Therefore, the LR can be defined as in Equation (1).

$$LR=\frac{p(E|H_p)}{p(E|H_d)} \quad (1)$$

In the case of FTC, the LR is the ratio between the two conditional probabilities of the measured difference (considered the evidence E) between the source-known texts (i.e., from the suspect) and the source-questioned texts (i.e., from the offender): one represents the probability of the evidence if they had been produced by the same author (H_p), and the other represents the probability of observing the same evidence if they had originated from different authors (H_d).

Thus, the evidence E, which is the measured difference between two texts (x,y) can be expressed as $\Delta(x,y)$. A bag-of-words model is used to represent each text in this study. Thus x and y stand for the vectors of relative word frequencies ($w_i^j,\ i \in \{1 \cdots N\}, j \in \{x,y\}$) of the texts to be compared ($x= \{w_1^x, w_2^x \cdots w_N^x\}$ and $y= \{w_1^y, w_2^y \cdots w_N^y\}$).

Thus, Equation (1) can be rewritten as Equation (2), where f denotes a probability density function.

$$LR=\frac{f(\Delta(x,y)|H_p)}{f(\Delta(x,y)|H_d)} = \frac{f(\Delta(\{w_1^x, w_2^x \cdots w_N^x\},\{w_1^y, w_2^y \cdots w_N^y\})|H_p)}{f(\Delta(\{w_1^x, w_2^x \cdots w_N^x\},\{w_1^y, w_2^y \cdots w_N^y\})|H_d)} \quad (2)$$

The probability density functions under H_p and H_d need to be trained from a data set of scores.

Once a forensic scientist has estimated the LR as the weight of the evidence, the LR is then interpreted as a multiplicative factor by which the Bayesian theorem is used to update the prior odds (the factfinder's prior beliefs about the hypotheses) to the posterior odds (the factfinder's beliefs after observing the evidence). The factfinder (e.g., jury or judge) is thus responsible for quantifying the prior odds of the hypotheses, and the forensic scientist is responsible for estimating the LR. That is, the ultimate decision of a case (i.e., guilty or not guilty) is determined by the factfinder, who must update the prior odds to the posterior odds with the LR.

In this study, LRs are estimated using a score-based approach that has been extensively studied with several evidence types (Bolck et al., 2015; Hepler et al., 2012; Ramos et al., 2017). An alternative to the score-based approach is the feature-based approach, which has been applied to authorship text evidence (Ishihara, 2014). In score-based approaches, the likelihood of the score—which is usually quantified as a similarity/difference or a distance between paired samples that can be represented in the form of feature vector—is assessed against the probabilistic distributions from the same-source and different-source scores. This process is called score-to-LR conversion. The conversion model must be constructed with relevant training data; naturally, the more the data, the more accurately the system can perform.

The types and conditions of the linguistic evidence used in criminal cases are all unique. It is often the case that relevant data for the case must be collected in a customised manner from scratch to train the score-to-LR conversion model. However, forensic scientists usually cannot afford to collect such a large number of data. Therefore, it is crucial that forensic scientists know how the FTC system's performance is influenced by the number of data. For this purpose, a series of experiments was conducted with the data that were synthesised by a Monte-Carlo simulation technique.

2 Experiment Design

Two sets of experiments were conducted, with the first set aiming to identify the conditions under which the FTC system optimally performs (see Section 3.1).

In the second set, with the best-performing conditions set, the FTC system's performance is assessed by altering the data number for training the score-to-LR conversion model (see Section 3.2). The database, pre-processing of data, logistic–regression calibration and assessment metrics are also discussed in this section.

2.1 Database

The current study used a portion of the Amazon Product Data Authorship Verification Corpus [1] (Halvani et al., 2017), which contained 21,534 product reviews from 3,228 reviewers. The review texts were equalised to be approximately 4kB in size, which corresponds to approximately 750 words in length. The reviewers contributed multiple product reviews for Amazon, but only those who produced six or more reviews were selected from the corpus, resulting in 2,160 reviewers. Only the first six reviews of each reviewer were selected for the two sets of experiments.

To compare a source-questioned (offender) sample and a source-known (suspect) sample, the six reviews were first separated into two groups: the first three and the last three, from which three documents that differed in word length (750, 1,500 and 2,250 words) were created by concatenating them. The first review text of each group was used as it originally appeared (i.e., as a document of 750 words). The first and second texts were also concatenated into a document of 1,500 words. All three texts were then combined into a document of 2,250 words. Documents of different word lengths were prepared for testing the correlation between the number of words and the system's performance.

2.2 Database Partition

The entire database was divided into the three mutually exclusive sub-databases of 'test', 'back-

[1] Available at http://bit.ly/1OjFRhJ.

ground' and 'development', each of which comprised documents from 720 authors (=2,160/3). The documents from the test database were used to assess the system's performance by generating same-author (SA) and different-author (DA) comparisons. From the 720 authors from the test database, each of whom had two documents for each word length, 720 SA comparisons and 517,680 DA comparisons (${}_{720}C_2 \times 2$) were possible for each word length.

The documents from the background database were used to obtain SA and DA scores, which were in turn used to train the score-to-LR conversion model. The composition of the background database was identical in quantity to that of the test database. That is, 720 SA scores and 517,680 DA scores could be obtained from the background database.

The resultant LRs after the score-to-LR conversion may not have been calibrated due to various reasons. In this case, the uncalibrated LRs had to be converted to interpretable LRs through a process of calibration. A typical and robust model for the calibration procedure is logistic regression (Morrison, 2013), and the development database was used to train the logistic regression. A more detailed explanation for logistic–regression calibration is provided in Section 2.4.

2.3 Tokenisation and a Bag-of-Words Model

Documents were tokenised with the tokens() function in the quanteda library (Benoit et al., 2018) of the R statistical package in the default setting. That is, all characters were changed to lower case and punctuation marks were *not* removed; the punctuation marks were thus considered single-word tokens. No stemming algorithm was applied.

The 420 most frequent words appearing in the entire dataset were selected as components for the bag-of-words model. The relative frequencies of the words in the model were then calculated for each document. These relative frequencies were used instead of word counts because the length of each document varied. The word frequencies of the bag-of-words vector were z-score normalised to equalise the amount of information across the words in the vector. If this step was not taken, then the information that was encoded in the frequently occurring words would substantially and unevenly influence the outcomes of the experiments, as word frequencies follow the distribution described by Zipf's law (Zipf, 1932).

2.4 Logistic–Regression Calibration

The LRs that are estimated using the score-based approach are usually well calibrated; they can thus be interpreted as the weight of evidence. As will be reported in Section 4, LRs become less calibrated when the background data are limited.

Figure 1 contains two Tippett plots which show the magnitude of the LRs derived from a simulation under a specific experimental condition (randomly generated scores from 10 authors for 2,250 words). Tippett plots show the cumulative proportion of the LRs of the SA comparisons, which are plotted rising from the left, as well as of the LRs of the DA comparisons, plotted rising from the right. For the Tippett plots, the cumulative proportion of trails is plotted on the Y-axis against the $\log_{10}$ LRs on the X-axis. The intersection of the two curves is the equal error rate (EER) which indicates the operating point at which the miss and false alarm rates are equal. As can be seen from Figure 1a, the intersection of the two curves is not aligned with $\log_{10}LR=0$. That means, the derived LRs are not well calibrated; thus they cannot be interpreted as the weight of evidence.

These uncalibrated LRs must be converted to calibrated LRs to be interpreted as the weight of evidence. A logistic–regression calibration (Brümmer and du Preez, 2006) is employed for this purpose. Logistic-regression calibration is operated by applying linear shifting and scaling to the uncalibrated LRs, in the log odds space, relative to a decision boundary; its aim is to minimise the magnitude and incidence of uncalibrated LRs that are known to misleadingly support the incorrect hypothesis, and also to maximise the values of uncalibrated LRs correctly supporting the hypotheses. A logistic-regression line, the weights of which are estimated on the basis of the LRs derived from a training database, is used to monotonically shift and scale the uncalibrated LRs to the calibrated LRs. By way of exemplification, assuming a logistic-regression line of the type $y=ax+b$ (where x is the uncalibrated LR and y is the calibrated LR, and the weights, a and b, are estimated on the basis of the (uncalibrated) LRs derived from the development database), the formula $y=ax+b$ is used to shift by the amount of b, and scale by the amount of a, the uncalibrated LRs to the calibrated LRs. The LRs presented in Figure 1b are the outcome of the application of logistic-regression calibration to the LRs given in Figure 1a.

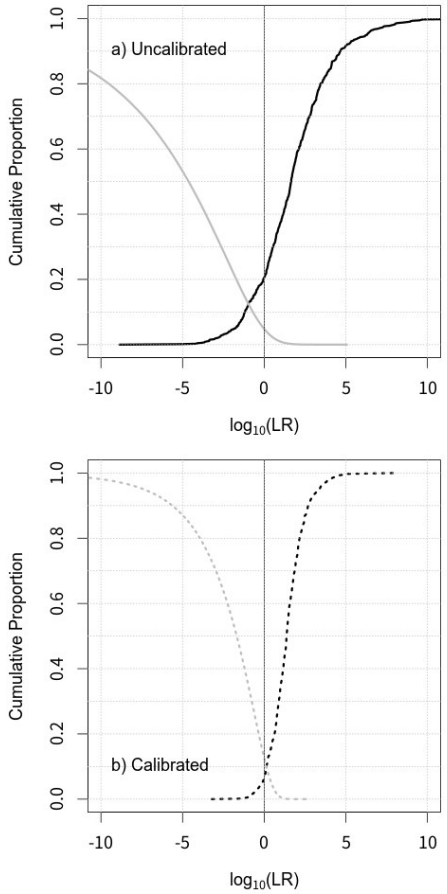


Figure 1: Example Tippett plots showing uncalibrated (Panel a) and calibrated (b) LRs. Black=SA LRs; Grey=DA LRs; Solid curves=uncalibrated LRs; Dotted curves=calibrated LRs.

2.5 Performance Evaluation

It is common to assess the performance of any identification or classification system based on its accuracy and error rates. However, accuracy and error rates are binary and categorical (e.g., correct or incorrect); this is not suitable for the nature of LR, which is gradient and continuous.

A more appropriate metric for assessing LR-based systems is arguably the log–LR cost (C_{llr}) (Brümmer and du Preez, 2006), which was originally developed for LR-based automatic speaker recognition systems. C_{llr} can be obtained through Equation (3).

$$C_{llr} = \frac{1}{2}\left(\begin{array}{l} \left[\frac{1}{N_{SA}}\sum_{i}^{N_{SA}} log_2\left(1+\frac{1}{LR_i}\right)\right] \\ +\left[\frac{1}{N_{DA}}\sum_{j}^{N_{DA}} log_2\left(1+LR_j\right)\right] \end{array} \right) \quad (3)$$

N_{SA} and N_{DA} refer to the number of SA and DA comparisons, respectively. LR_i and LR_j refer to the linear LRs that are derived from these SA and DA comparisons. In this metric, all LRs (except ±infinity) are attributed penalties in proportion to their magnitudes, with the LRs that support the counter-factual hypotheses being more severely penalised. The C_{llr} is based on information theory, and if the C_{llr} value is higher than one, then the system is performing worse than not utilising the evidence at all.

The C_{llr} is a metric that assesses a system's overall validity. It comprises two components: discrimination loss (C_{llr}^{min}) and calibration loss (C_{llr}^{cal}): $C_{llr}=C_{llr}^{min}+C_{llr}^{cal}$. The C_{llr}^{min} is a theoretical minimum C_{llr} value that can be obtained through pool adjacent violators algorithms (Brümmer and du Preez, 2006).

3 Experiments

3.1 Preparatory Experiments and Outcomes

A series of FTC experiments was conducted with a score-based LR approach to identify under what conditions the system would yield the best outcome. In these experiments, scores were measured with Cosine distance, with the bag-of-words model consisting of N most frequent words. The scores were then converted to their LRs based on the conversion model that was trained by the scores calculated from the SA and DA comparisons, which were compiled from the background database. The size (N) of the bag-of-words vector is incremented from N=20 to N=420 by 20 to identify the best-performing N. The Normal, Log-Normal, Weibull and Gamma models were tried as possible conversion models, but only the model that fit the data best in terms of the Akaike information criterion (AIC) (Akaike, 1974) was selected for each experiment (separately for the SA and DA models). Cosine distance was used because of its superior performance to other measures (Evert et al., 2017; Smith and Aldridge, 2011).

The C_{llr} values are plotted as a function of the feature number (N) in Figure 2, separately for 750, 1,500 and 2,250 words. Regardless of the word length, the system performed best with N=260. The overall trend for the C_{llr} trajectory is similar across the word lengths, revealing a relatively large improvement in performance as the N increased from 20 to 120 and the C_{llr} values started converging towards N=260. After N=260, the performance remained relatively unchanged, indicating that the inclusion of less-frequent words did not contribute to the improvement.

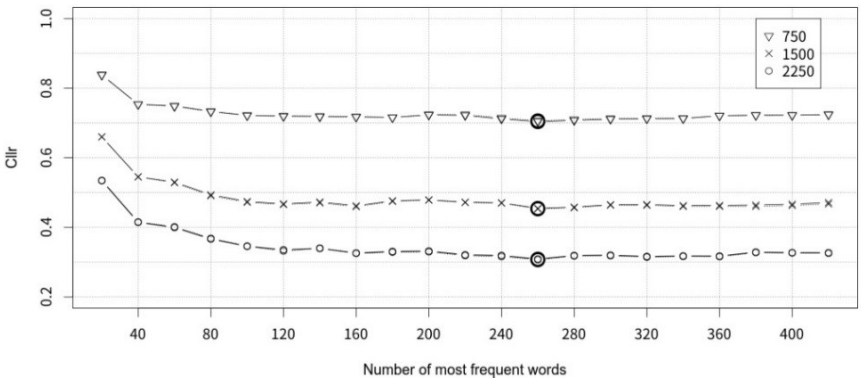


Figure 2: C_{llr} values plotted as a function of the number of features, separately for the word lengths of 750, 1,500 and 2,250. The large circles indicate the best C_{llr}.

The best-fitted models when N=260 are outlined in Table 1 and are used for the Monte-Carlo simulation.

	SA scores	DA scores
750	Weibull	Weibull
1500	Weibull	Normal
2250	Weibull	Normal

Table 1: Best-fitted parametric models for the SA and DA scores.

3.2 Experiments with the Monte-Carlo Simulation

In the preparatory experiment, the score-to-LR conversion models were trained with the data in the background database, which comprised texts written by 720 authors. Using the model as the basis, the scores of X number of authors (X=[5, 10, 20, 30, 40, 60, 80, …, 720]) were randomly generated 20 times to build the conversion models.

The Normal, Log-Normal, Weibull and Gamma parametric models were fitted to the scores that were randomly generated separately for SA and DA comparisons in the maximum likelihood estimation method. The best-fitted model was chosen according to its AIC values.

Figure 3 illustrates the simulation process for the length of 750 words. Out of the texts written by 720 authors from the background database, 720 SA and 517,680 DA scores were estimated. These scores are plotted as histograms: the white histogram represents SA and the grey histogram represents DA. Their fitted models (Weibull) are presented as solid red and blue curves, respectively. From these two models, the scores for the SA and DA comparisons—which are possible from 30 authors (i.e., 30 SA and 870 SA scores)—were randomly generated 20 times. Their models are represented by thin black curves. These models were used for the score-to-LR conversion.

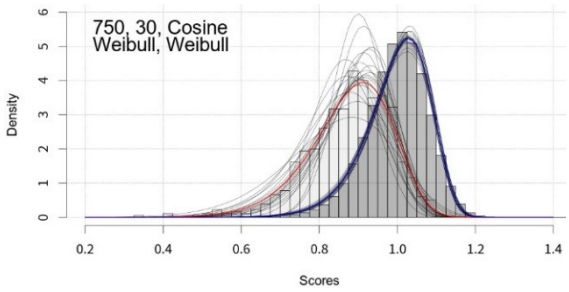


Figure 3: Illustration of a Monte-Carlo simulation with the base SA and DA scores, of which the histograms are white and grey, respectively. The red and blue curves are models of the SA and DA scores, respectively. The thin lines represent the models of the 20 sets of randomly generated scores from 30 authors.

4 Results and Discussions

The boxplots presented in Figure 4 reveal the degree of fluctuations in the C_{llr} values of the 20 simulations; they also indicate how the C_{llr} values converge as the number of authors increases.

Regardless of the word length, the FTC system's performance substantially fluctuates when the background database only comprises the text samples from 5~10 authors; that is, the performance is not stable. However, this instability quickly recovers if the text samples are collected from 20 or more authors. This is a positive finding in terms of FTC's practical application, as forensic scientists cannot afford the time and money required to collect a large number of data that are relevant to each case if they cannot find an already-existing dataset that is suitable to the case.

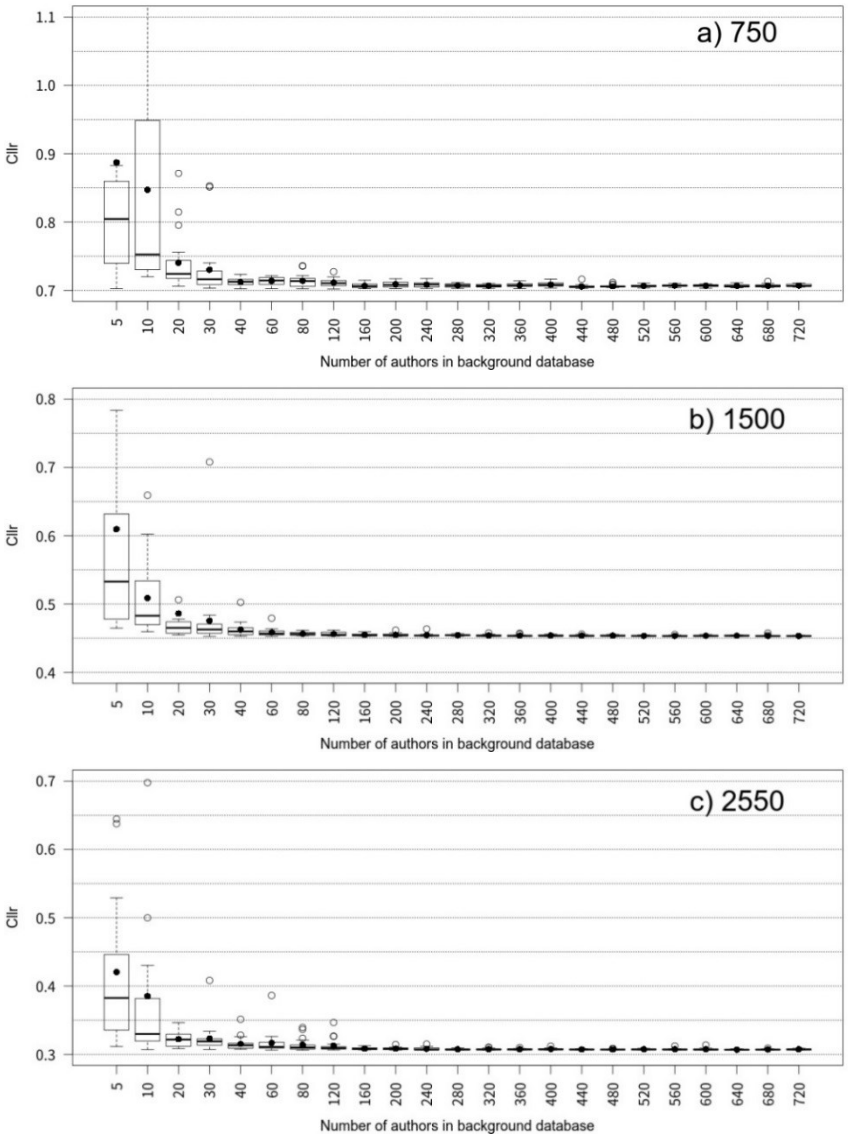


Figure 4: Boxplots displaying the degree of fluctuation in C_{llr} values as a function of the size of the background database. Black circles indicate the mean C_{llr} values for each size of the background database.

It is evident from Figure 4 (black circles) that the system's overall performance improves exponentially from N=5 to N=40, resulting in the outcome in which the performance with N=40 is nearly compatible with its performance with N=720.

To further investigate the reasons underlying the fluctuations in performance (especially with the small number of N), the C_{llr}^{min} and C_{llr}^{cal} values (discrimination loss and calibration loss, respectively) are plotted separately in Figures 5 and 6, respectively. They are presented in the same manner as Figure 4. As can be observed in Figure 5, being apart from the word length of 750, the system's discriminability is highly stable, even with small Ns. Specifically, regarding the word length of 2,250, Figure 5c reveals that the C_{llr}^{min} values are constant and far less fluctuated, as they are not affected by the number of authors in the background database. That is, in terms of discrimination performance, when many words (e.g., 1,500 and 2,550 words) are available, the system is robust and stable against a small background population size.

In contrast, Figure 6 indicates that the C_{llr}^{cal} values exhibit a highly similar trend to that of the C_{llr} values that are plotted in Figure 4—in that, a great variability in the C_{llr}^{cal} values is observed when the number of authors is small (e.g., N=5~10); however, this variability begins converging rapidly with more authors. This signifies that the C_{llr}^{cal} values also demonstrate a quick recovery with more authors. The observations drawn from Figures 5 and 6 reveal that the poor performance associated with a small number of authors (N=5~10), as indicated by the C_{llr} values from Figure 4, is not due to the system's poor discriminability, but due to poor calibration.

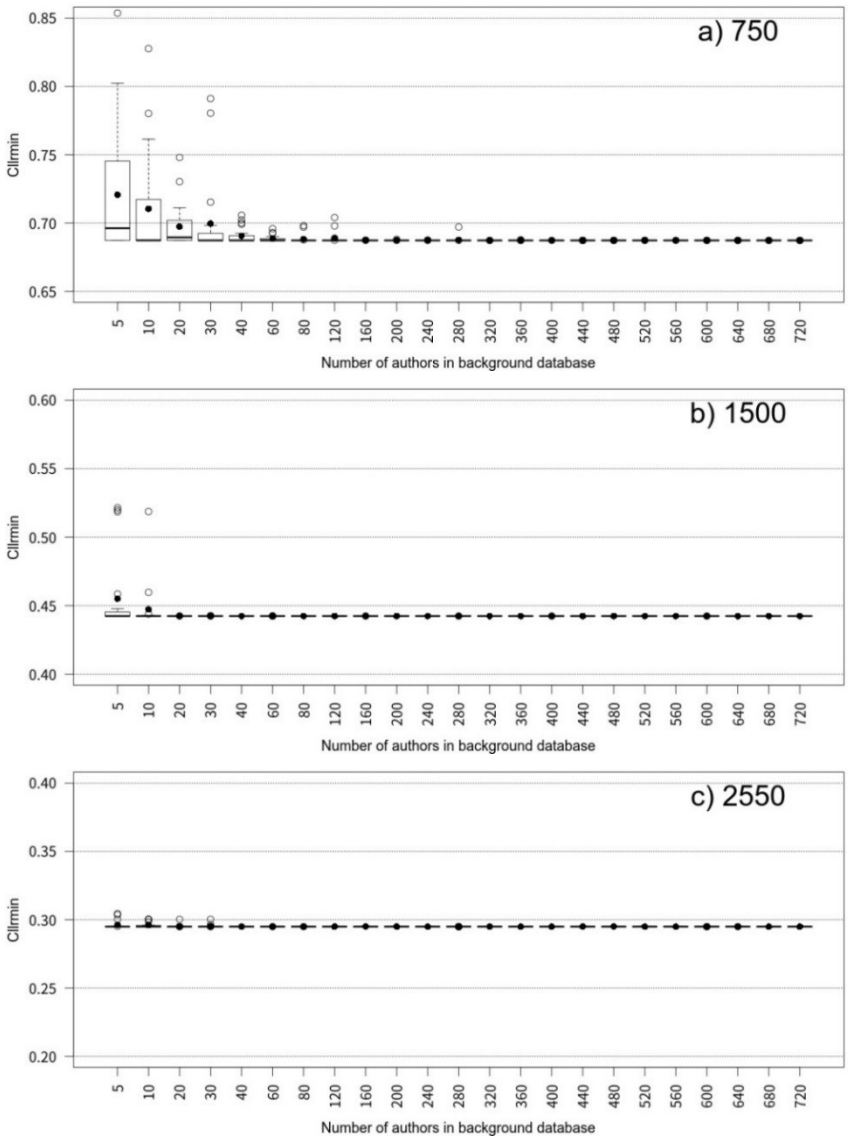


Figure 5: Boxplots displaying the degree of fluctuation in C_{llr}^{min} values as a function of the size of the background database. Black circles indicate the mean C_{llr}^{min} values for each size of the background database.

Following this interpretation, logistic–regression calibration was applied to all LRs, in which a gain in overall performance was expected. The C_{llr} values of the calibrated LRs are again plotted as boxplots in Figure 7. It is apparent from Figure 7 that the system's performance has noticeably improved in both stability and accuracy; the degree of fluctuations in the C_{llr} values is lessened and the mean C_{llr} values are lower, even with small *N*s.

Ishihara (2016) previously investigated how background population size affected the performance of an LR-based FTC system. In the experiments, the LRs were estimated using the multivariate kernel density (MVKD) LR formula (Aitken and Lucy, 2004), with two to eight stylometric features. Texts collected from 140 authors were used to extract necessary statistical information for a Monte-Carlo simulation, for which a mixture Gaussian model was used. The MVKD is a type of feature-based approach for estimating LRs. The population size was incremented by 10 from 10 authors to 140 authors.

Although a direct comparison between the current study and Ishihara's (2016) study cannot be validly made, some noticeable differences can still be highlighted. The number of features (2~8) used in Ishihara's study was far smaller than that of the current study (260), and Ishihara reported a great improvement in C_{llr} (from 10 to 50~60 authors), after which a small but continuous improvement could be observed with more authors. He also reported a relatively high variability in C_{llr}, even with a large number of authors (e.g., 140).

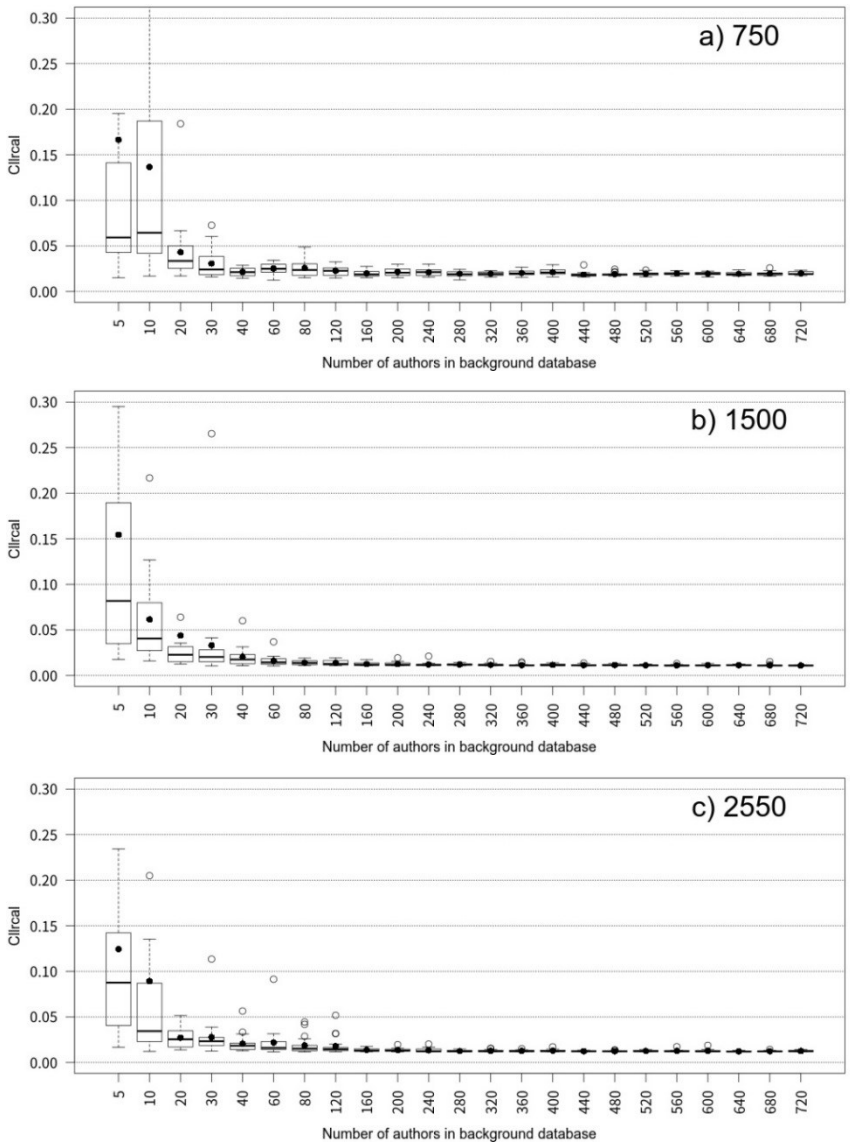


Figure 6: Boxplots showing the degree of fluctuation in C_{llr}^{cal} as a function of the size of the background database. Black circles indicate the mean C_{llr}^{cal} values for each size of the background database.

In light of these comparative observations, the FTC system's performance appears to reach its optimum with a smaller population size for the score-based approach rather than for the feature-based approach. Further, the fluctuation in performance also begins converging with a lesser number of background data for the score-based approach than for the feature-based approach. The relative robustness of the score-based approach that the current study revealed for linguistic text evidence aligns with the findings in previous studies regarding other types of evidence (Aitken, 2018; Bolck et al., 2015). However, the difference in performance between the score- and feature-based approaches must be further investigated under mutually comparable conditions.

Based on Figure 7, it can be concluded that logistic–regression calibration leads to an improvement in terms of the system's stability and validity. For training the logistic–regression weights, the development database that comprised the texts from 720 authors was employed. It is evident that the calibration performance also mainly relies on the quantity of the data in the development database. The positive outcome after applying the calibration is likely attributable to the amount of data in the development database. Therefore, it is pertinent to analyse how the development database's size influences the FTC system's performance, as the application of calibration appears to be essential when the background database is substantially small in number (e.g., 5~10 authors).

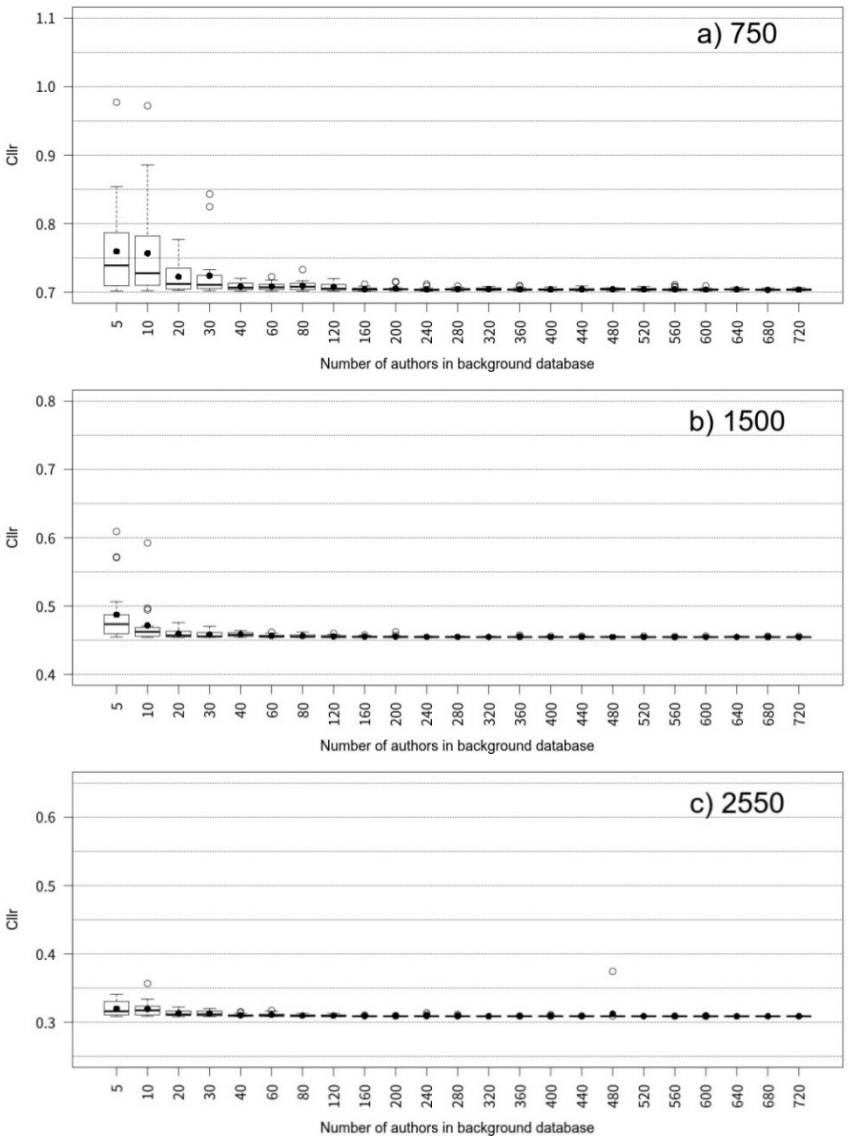


Figure 7: Boxplots revealing the fluctuation of C_{llr} after logistic–regression calibration.

5 Conclusion and Further Study

The robustness and stability of a score-based LR FTC system with a bag-of-words model were investigated with different numbers of background population data, which were synthesised by a Monte-Carol simulation. The experiments' results revealed that the score-based FTC system is fairly robust and stable in performance against the limited number of background population data. For example, with 40~60 authors, the performance is both nearly compatible and as stable as with 720 authors. This is a beneficial finding for FTC practitioners. Additionally, the instability and suboptimal performance observed in terms of C_{llr} with a small number of data (e.g., 5~20 authors) were mainly attributed to poor calibration (i.e., the derived LRs were not calibrated) rather than to the poor discriminability potential.

A comparison with the outcomes of previous studies indicates that the score-based approach may be more robust against a limited number of background population data than a feature-based approach; however, this point warrants further study.

Acknowledgements

The author thanks the reviewers for their valuable comments.

References

Aitken, C. G. G. (2018) Bayesian hierarchical random effects models in forensic science. *Frontier in Genetics* *9*(Article 126): 1-14. https://doi.org/10.3389/fgene.2018.00126

Aitken, C. G. G. and Lucy, D. (2004) Evaluation of trace evidence in the form of multivariate data. *Journal of the Royal Statistical Society, Series C (Applied Statistics)* *53*(1): 109-122. https://dx.doi.org/10.1046/j.0035-9254.2003.05271.x

Aitken, C. G. G. and Stoney, D. A. (1991) *The Use of Statistics in Forensic Science*. New York: Ellis Horwood.

Aitken, C. G. G. and Taroni, F. (2004) *Statistics and the Evaluation of Evidence for Forensic Scientists*. Chichester: John Wiley & Sons.

Akaike, H. (1974) A new look at the statistical model identification. *IEEE Transactions on Automatic Control* *19*(6): 716-723. https://dx.doi.org/10.1109/TAC.1974.1100705

Balding, D. J. (2005) *Weight-of-Evidence for Forensic DNA Profiles*. Hoboken: John Wiley & Sons.

Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S. and Matsuo, A. (2018) quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software* *3*(30): 774-776. https://doi.org/10.21105/joss.00774

Bolck, A., Ni, H. F. and Lopatka, M. (2015) Evaluating score- and feature-based likelihood ratio models for multivariate continuous data: Applied to forensic MDMA comparison. *Law, Probability and Risk* *14*(3): 243-266. https://dx.doi.org/10.1093/lpr/mgv009

Brümmer, N. and du Preez, J. (2006) Application-independent evaluation of speaker detection. *Computer Speech and Language* *20*(2-3): 230-275. https://dx.doi.org/10.1016/j.csl.2005.08.001

Evert, S., Proisl, T., Jannidis, F., Reger, I., Pielström, S., Schöch, C. and Vitt, T. (2017) Understanding and explaining Delta measures for authorship attribution. *Digital Scholarship in the Humanities* *32*(suppl_2): ii4-ii16. https://doi.org/10.1093/llc/fqx023

Evett, I. W., Lambert, J. A. and Buckleton, J. S. (1998) A Bayesian approach to interpreting footwear marks in forensic casework. *Science & Justice* *38*(4): 241-247. https://dx.doi.org/10.1016/S1355-0306(98)72118-5

Good, I. J. (1991) Weight of evidence and the Bayesian likelihood ratio. In C. G. G. Aitken and D. A. Stoney (eds.), *The Use of Statistics in Forensic Science* 85-106. Chichester: Ellis Horwood.

Halvani, O., Winter, C. and Graner, L. (2017). Authorship verification based on compression-models. *arXiv preprint arXiv:1706.00516*. Retrieved on 25 June 2020 from http://arxiv.org/abs/1706.00516

Hepler, A. B., Saunders, C. P., Davis, L. J. and Buscaglia, J. (2012) Score-based likelihood ratios for handwriting evidence. *Forensic Science International* *219*(1-3): 129-140. http://dx.doi.org/10.1016/j.forsciint.2011.12.009

Ishihara, S. (2014) A likelihood ratio-based evaluation of strength of authorship attribution evidence in SMS messages using N-grams. *International Journal of Speech Language and the Law 21*(1): 23-50. http://dx.doi.org/10.1558/ijsll.v21i1.23

Ishihara, S. (2016) An effect of background population sample size on the performance of a likelihood ratio-based forensic text comparison system: A Monte Carlo simulation with Gaussian mixture model. In T. Cohn (ed.), *Proceedings of Proceedings of the Australasian Language Technology Association Workshop 2016*: 113-121.

Ishihara, S. (2017a) Strength of forensic text comparison evidence from stylometric features: A multivariate likelihood ratio-based analysis. *The International Journal of Speech, Language and the Law* *24*(1): 67-98. https://doi.org/10.1558/ijsll.30305

Ishihara, S. (2017b) Strength of linguistic text evidence: A fused forensic text comparison system. *Forensic Science International 278*: 184-197. https://doi.org/10.1016/j.forsciint.2017.06.040

Lund, S. P. and Iyer, H. (2017) Likelihood ratio as weight of forensic evidence: A closer look. *Journal of Research of the National Institute of Standards and Technology* *122*(Article 27): 1-32. https://doi.org/10.6028/jres.122.027

Marquis, R., Bozza, S., Schmittbuhl, M. and Taroni, F. (2011) Handwriting evidence evaluation based on the shape of characters: Application of multivariate likelihood ratios. *Journal of Forensic Sciences* *56*(Suppl_1): S238-242. https://dx.doi.org/10.1111/j.1556-4029.2010.01602.x

Morrison, G. S. (2009) Forensic voice comparison and the paradigm shift. *Science & Justice 49*(4): 298-308. https://dx.doi.org/10.1016/j.scijus.2009.09.002

Morrison, G. S. (2013) Tutorial on logistic-regression calibration and fusion: Converting a score to a likelihood ratio. *Australian Journal of Forensic Sciences* *45*(2): 173-197. https://dx.doi.org/10.1080/00450618.2012.733025

Neumann, C., Champod, C., Puch-Solis, R., Egli, N., Anthonioz, A. and Bromage-Griffiths, A. (2007) Computation of likelihood ratios in fingerprint identification for configurations of any number of minutiae. *Journal of Forensic Science 52*(1): 54-64. https://dx.doi.org/10.1111/j.1556-4029.2006.00327.x

Ramos, D., Krish, R. P., Fierrez, J. and Meuwly, D. (2017) From biometric scores to forensic likelihood ratios. In M. Tistarelli and C. Champod (eds.), *Handbook of Biometrics for Forensic Science* 305-327. Cham: Springer.

Robertson, B., Vignaux, G. A. and Berger, C. E. H. (2016) *Interpreting Evidence: Evaluating Forensic Science in the Courtroom* (2nd ed.). Chichester: John Wiley and Sons, Inc.

Smith, P. W. H. and Aldridge, W. (2011) Improving authorship attribution: Optimizing Burrows' Delta method. *Journal of Quantitative Linguistics 18*(1): 63-88. https://dx.doi.org/10.1080/09296174.2011.533591

Zipf, G. K. (1932) *Selected Studies of the Principle of Relative Frequency in Language*. Cambridge: Harvard University Press.

Feature-Based Forensic Text Comparison Using a Poisson Model for Likelihood Ratio Estimation

Michael Carne and Shunichi Ishihara
Speech and Language Laboratory
The Australian National University
michael.carne@anu.edu.au, shunichi.ishihara@anu.edu.au

Abstract

Score- and feature-based methods are the two main ones for estimating a forensic likelihood ratio (LR) quantifying the strength of evidence. In this forensic text comparison (FTC) study, a score-based method using the Cosine distance is compared with a feature-based method built on a Poisson model with texts collected from 2,157 authors. Distance measures (e.g. Burrows's Delta, Cosine distance) are a standard tool in authorship attribution studies. Thus, the implementation of a score-based method using a distance measure is naturally the first step for estimating LRs for textual evidence. However, textual data often violates the statistical assumptions underlying distance-based models. Furthermore, such models only assess the similarity, not the typicality, of the objects (i.e. documents) under comparison. A Poisson model is theoretically more appropriate than distance-based measures for authorship attribution, but it has never been tested with linguistic text evidence within the LR framework. The log-LR cost (C_{llr}) was used to assess the performance of the two methods. This study demonstrates that: (1) the feature-based method outperforms the score-based method by a C_{llr} value of ca. 0.09 under the best-performing settings and; (2) the performance of the feature-based method can be further improved by feature selection.

1 Introduction

The essential part of any source-detection task is to assess the similarity or difference between the objects or items under comparison. For this purpose, in stylometric studies too, various distance measures have been devised and tested, particularly in studies concerned with the authorship of text sources (Argamon, 2008; Burrows, 2002; Hoover, 2004a; Smith and Aldridge, 2011). Burrows's Delta (Burrows, 2002) is probably the most studied distance measure in stylometric studies, and its effectiveness and robustness have been demonstrated for a variety of texts from different genres and languages (AbdulRazzaq and Mustafa, 2014; Hoover, 2004b; Rybicki and Eder, 2011; Þorgeirsson, 2018). Since Burrows (2002), several variants, including, for example, those based on Euclidian distance, Cosine similarity and Mahalanobis distance have been proposed to better deal with the unique characteristics of linguistic texts, expecting to result in a better identification and discrimination performance (Argamon, 2008; Eder, 2015; Hoover, 2004b; Smith and Aldridge, 2011).

Similarity- and distance-based measures make some assumptions about the distribution of the underlying data. For example, a Laplace distribution is assumed by Burrows's Delta, which itself is based on Manhattan distance, and a normal distribution by the Euclidean and cosine distances. However, it is well known that stylometric features do not always conform to, for example, a normal distribution (Argamon, 2008; Jannidis et al., 2015). Moreover, a normal distribution is not theoretically appropriate for discrete count data (e.g. occurrences of function words) Figure 1 shows the distributions of the counts of three words ('a', 'not' and 'they'), sampled from the database used in the current study. Frequently-occurring words, such as 'a' (Figure 1a), tend to be normally distributed. However the distribution starts skewing positively for less-frequently-occurring words, such as 'not' (Figure 1b) and 'they' (Figure 1c). In order to fill this gap between the theoretical assumption arising from distance measures and the nature of textual data, a one-level Poisson model is used in this study.

In the 1990s, the success of DNA analysis and some important United States court rulings, estab-

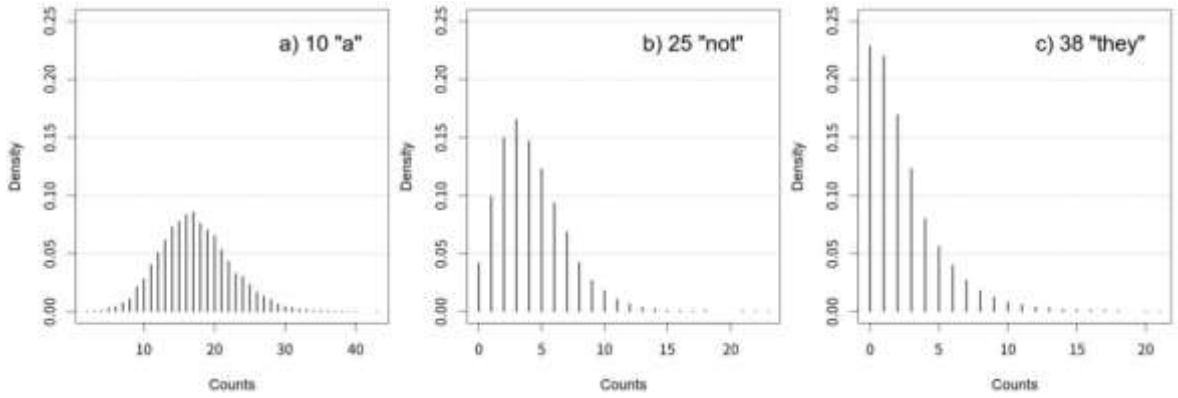


Figure 1: Histograms showing the distributional patterns of the counts of three words from the database; 'a', 'not' and 'they' for Panel a), b) and c), respectively. They are the 10th, 25th and 38th most frequently-occurring words in the database.

lishing the standard for expert evidence to be admitted in court, promoted the likelihood ratio (LR)-based approach as the standard for evaluating and presenting forensic evidence in court (Association of Forensic Science Providers, 2009). Although it is far less extensively studied than other areas of forensic science, it has been demonstrated that the LR framework can be applied successfully to linguistic textual evidence (Ishihara, 2014, 2017a, 2017b).

1.1 Previous Studies

There are two methods for deriving an LR model for forensic data, score- and feature-based. Each method has different strengths and shortcomings. The use of score-based methods is prevalent across different types of forensic evidence due to its robustness and ease of implementation relative to feature-based methods. The advantages and disadvantages of the methods are explained in §3.3 and §3.4.

Almost all previous LR studies, both feature- and score-based, use continuous data for LR estimation. Studies using feature-based LR models derived from probability distributions appropriate for discrete (or categorical) forensic features are rare.

To the best of our knowledge, Aitken and Gold (2013) and Bolck and Stamouli (2017) are the only two existing studies of this kind within the LR framework. Aitken and Gold (2013) propose a univariate discrete model for estimating LRs. They conducted only a small-scale experiment using limited data and features, which were used mainly for explanatory purposes.

Bolck and Stamouli (2017) investigate discrete multivariate models for estimating LRs using categorical data from gunshot residue. This study however uses a relatively low-dimensional feature space (only 12 features), and its modelling approach assumes independence between features. Text evidence however usually involves high-dimensional vector spaces and independence cannot be assumed, given correlation between features. The present study seeks to investigate these challenges in LR-based forensic text comparison (FTC) using discrete textual data in the form of counts of the *N* most frequently occurring words. It implements a feature-based LR model derived from the Poisson distribution, with logistic-regression fusion and calibration used as a means for dealing with correlation between features. This approach is compared to a score-based method using the cosine distance. To the best of our knowledge, this is the first FTC study to trial a feature-based method with a Poisson model in the LR framework.

2 Likelihood Ratio Framework

The LR framework has been proposed as a means of quantifying the weight of evidence for a variety of forensic evidence, including DNA (Evett and Weir, 1998), voice (Morrison et al., 2018; Rose, 2002), finger prints (Neumann et al., 2007), handwriting (Chen et al., 2018; Hepler et al., 2012), hair strands (Hoffmann, 1991), MDMA tablets (Bolck et al., 2009), evaporated gasoline residual (Vergeer et al., 2014) and earmarks (Champod et al., 2001). Collected forensic items from known- (e.g. a suspect's known text samples) and questioned-source (e.g. text samples from the offender) can be evaluated by estimating the LR under two competing hypotheses. One specifying the prosecution (or the same-author) hypothesis (H_p), and the other the defence (or the different-author) hypothesis (H_d).

These are expressed as a ratio of condition probabilities as shown in Equation 1).

$$LR = \frac{f(x, y|H_p)}{f(x, y|H_d)} \quad 1)$$

where x and y are feature values obtained from the known-source and questioned-source respectively. The relative strength of the evidence with respect to the competing hypotheses is reflected in the magnitude of the LR. The more the LR deviates from unity (LR = 1), the greater support for either the H_p (LR > 1) or the H_d (LR < 1).

The LR is concerned with the probability of evidence, given the hypothesis (either prosecution or defence), which is in concordance with the role of an expert witness in court, leaving the trier-of-fact to be concerned with the probability of either hypothesis, given the evidence.

3 Experiments

The two main approaches for estimating LRs, namely the score- and feature-based methods, will be implemented and their performance compared. After the database (§3.1) and the pre-processing and modelling techniques (§3.2) are introduced, the two methods are explained in §3.3 and §3.4, respectively, along with their pros and cons. Fusion/calibration techniques and performance metrics are described in §3.5 and §3.6, respectively.

3.1 Database

Data for the experiments were systematically selected from the Amazon Product Data Authorship Verification Corpus[1] (Halvani et al., 2017), which contains 21,534 product reviews posted by 3,228 reviewers on Amazon. Many of the reviewers contributed six or more reviews on different topics. Sizes of review texts are equalised to ca. 4 kB, which corresponds to approximately 700 words in length. From the corpus, the authors (= reviewers) who contributed more than six reviews longer than 700 words, were selected as the database for simulating offender vs. suspect comparisons. We decided on six reviews to maximise the number of same-author comparisons possible from the database. This resulted in 2,157 reviewers and a database containing a total of 12,942 review texts. Each review was further equalised to 700 words. The first three reviews of each author were grouped as source-known documents (i.e. suspect documents) and the second three reviews were grouped as source-unknown documents (i.e. offender documents). The total number of word tokens in each group was 2,100, which constitutes a realistic sample size for forensic studies in our casework experience. The database was evenly divided into three mutually exclusive test, background and development sub-databases, each consisting of documents from 719 authors.

The documents stored in the test database were used for assessing the FTC system performance by simulating same-author (SA) and different-author (DA) comparisons. From the 719 authors in the test database, 719 SA comparisons and 516,242 (= ${}_{719}C_2 \times 2$) DA comparisons can be simulated.

The documents stored in the background database were used differently depending on the method. For the score-based method, they were used to train the score-to-LR conversion model, and in the feature-based method, they were used to assess the typicality of the documents under comparison.

For various reasons, including violation of modelling assumptions and data scarcity, the estimated LRs may not be well calibrated, in which case they cannot be interpreted as the strength of evidence (Morrison, 2013). A development database is typically used to calibrate the raw LRs via logistic-regression. However, in this study it was found that the LRs derived from the score-based method were well calibrated to begin with; thus logistic-regression calibration was not required. The development database was only used to fuse and calibrate the LRs derived from the feature-based method in this study. A more detailed explanation on logistic regression fusion/calibration is given in §3.5.

The type of communication that the current study focuses on is the one-to-many type of communication. Although the selected database is designed specifically for authorship verification tests, it is not a forensic database. To the best of our knowledge, there are no databases available of real forensic messages, nor any specifically designed with forensic conditions in mind. Nevertheless, the database used in this study was judged to be the most appropriate of existing databases to simulate a forensic scenario involving one-to-many communication. The product reviews were written as personal opinions and assessments of a given product addressing a public audience, and the review

[1] http://bit.ly/1OjFRhJ

messages have a clear purpose; conveying one's views to others. So, the content of the messages is focused and topic specific, like the malicious use of the one-to-many type of communication platforms (e.g. the spread of fake news, malicious intent and the defamation of individuals/organisations).

3.2 Tokenisation and Bag of Words Model

The `tokens()` function from the `quanteda` library (Benoit et al., 2018) in `R` (R Core Team, 2017) was used to tokenise the texts with the default settings. That is, all characters were converted to lower case without punctuation marks being removed; punctuation marks are treated as single word tokens. In order to preserve individuating information in author's morpho-syntactic choices (HaCohen-Kerner et al., 2018; Omar and Hamouda, 2020), no stemming algorithm was applied.

The 400 most frequent occurring words in the entire dataset were selected as components for a bag-of-words model. The occurrences of these words were then counted for each document. More specifically, the documents (x, y) under comparison were modelled as the vectors ($x = \{w_1^x, w_2^x \cdots w_N^x\}$ and y = $\{w_1^y, w_2^y \cdots w_N^y\}$) with the word counts ($w_i^j, i \in \{1 \cdots N\}, j \in \{x, y\}$).

In the experiments, the size (N) of the bag-of-words vector is incremented by 5 from $N = 5$ to $N = 20$, and then by 20 until $N = 400$. The 400 most frequent words are sorted according to their frequencies in a descending order. $N = 400$ was chosen as the cap of the experiments because the experimental results showed the performance ceiling before $N = 400$.

3.3 Score-based Method with Distance Measure (Baseline Model)

Estimating LRs using score-based methods is common in the forensic sciences (Bolck et al., 2015; Chen et al., 2018; Garton et al., 2020; Morrison and Enzinger, 2018). For score-based methods, the evidence consists of scores, $\Delta(x, y)$, which are often measured as the distance between the suspect and offender samples. In this case, the LR can be estimated as the ratio of the two probability densities of the scores under the two competing hypothesis as given in Equation 2).

$$LR = \frac{f(x, y|H_p)}{f(x, y|H_d)} = \frac{f(\Delta(x, y)|H_p)}{f(\Delta(x, y)|H_d)} \qquad 2)$$

The probability densities are trained on the scores obtained from the SA and DA comparisons generated from a background database. That is, the probability densities are used as a score-to-LR conversion model. The Cosine distance was used as a baseline in the current study as its superior performance has been previously reported in authorship attribution studies (Evert et al., 2017; Smith and Aldridge, 2011). The three documents from each group were concatenated as a document of 2,100 words for the score-based method. The count of each word was *z*-score normalised in order to avoid the most frequent words biasing the estimation of the LRs. The *z*-score normalised values were used to represent each document in the bag-of-words model described in §3.2.

Score-based methods project the complex, multivariate feature vector into a univariate score space (Morrison and Enzinger, 2018: 47). Its robustness and ease of implementation for various types of forensic evidence have been reported as benefits (Bolck et al., 2015). However, information loss is inevitable due to the reduction in dimensionality. Another shortcoming is that score-based methods do not account for the typicality of the evidence. Because of these shortcomings, it is reported that the magnitude of the derived LRs is generally weak (Bolck et al., 2015; Morrison and Enzinger, 2018). Nevertheless, the approach has been widely studied across a variety of forensic evidence.

3.4 Feature-based Method with Poisson Model

Feature-based methods maintain the multivariate structure of the data through estimation of the LR directly from the feature values (Bolck et al., 2015). This has the potential to prevent information loss but comes at the cost of added model complexity and reduced computational efficiency. Feature-based methods allow the typicality, not only the similarity, of forensic data to be assessed. In feature-based methods, the LR is estimated as a ratio of two conditional probabilities, which express the similarity and typicality of the samples under comparison. These correspond respectively to the numerator and denominator of Equation 1). Similarity, in this context, refers to how similar/different the source-known and source-questioned documents are with respect to their measured properties, and typicality means how typical/atypical they are in the relevant population. In this study a Poisson distribution was used to construct the LR

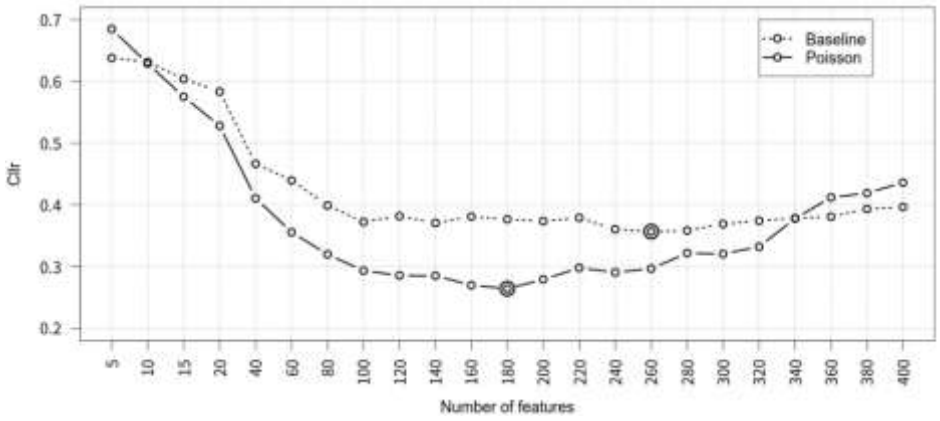


Figure 2: The C_{llr} values of the LRs with the N number of features indicated in the y-axis are plotted separately for the Baseline and the Poisson models. The features are sorted according to the frequencies of the words. The large circles indicate the best C_{llr} values for the models.

model. The probability mass function for the Poisson distribution is given in Equation 3) and the LR model in Equation 4).

$$p(k;\lambda) = e^{-\lambda}\frac{\lambda^k}{k!} \qquad 3)$$

In Equation 3), λ is the shape parameter which indicates the average number of events in the given time interval or space. That is, letting $x = (x_{1,} \cdots x_k)$ and $y = (y_{1,} \cdots y_k)$ be the counts of a given word for the suspect and offender documents, an LR for the pair of documents is estimated for the word by Equation 4).

$$LR = \frac{f(x,y|H_p)}{f(x,y|H_d)} = \frac{f(y|x,H_p)}{f(y|H_d)} = \frac{f(y|\lambda_x)}{f(y|\lambda_B)} = \frac{\prod_{i=1}^{k}(y_i|\lambda_x)}{\prod_{i=1}^{k}(y_i|\lambda_B)} = \frac{\prod_{i=1}^{k} e^{-\lambda_x}\frac{\lambda_x^{y_i}}{y_i!}}{\prod_{i=1}^{k} e^{-\lambda_B}\frac{\lambda_B^{y_i}}{y_i!}} \qquad 4)$$

where the λ_x is the mean of x and the λ_B is the overall mean λ of the background database. Both the suspect and offender documents consist of three texts; thus $k = 3$. The second fraction of Equation 4) can be reduced to the third fraction by assuming that the probability of the feature values x is independent of whether x comes from the same source as y or not, and that x and y are independent if H_d is true. LRs were estimated separately for each of the 400 features.

3.5 Logistic-Regression Fusion and Calibration

If the LRs derived separately for the 400 features were independent of one another, they could be multiplied in a naïve Bayesian manner for an overall LR. However, it is known empirically that independence cannot be assumed (Argamon, 2008; Evert et al., 2017). This means, they need to be fused instead, taking the correlations into consideration. Fusion enables us to combine and calibrate multiple parallel sets of LRs from different sets of features/models or even different forensic detection systems, with the output being calibrated LRs. Logistic-regression fusion/calibration (Brümmer and du Preez, 2006) is a commonly used method for LR-based systems. A logistic-regression weight needs to be calculated for each set of LRs, as shown in Equation 5).

$$Fused\ LR = a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n + b \qquad 5)$$

where, $x_{1,}\ x_2,\ x_3 \ldots x_n$ are the LRs of the first through nth set, and $a_{1,}\ a_2,\ a_3 \ldots a_n$ are the corresponding logistic-regression weights for scaling. The logistic-regression weight for shifting is b. The weights are obtained from the LRs estimated for the SA and DA comparisons from documents in the development database. The number (N) of features to be fused were incremented by 5 from $N = 5$ to $N = 20$, and then by 20 until $N = 400$.

The same technique can be applied to a single set of LRs, in which case, logistic-regression is used only for calibration. However, it was not applied to the LRs derived with the score-based method as they were well-calibrated to start with.

3.6 Evaluation Metrics: Log-LR Cost

The log-LR cost (C_{llr}), which is a gradient metric based on LR, was used to assess the performance of the FTC systems for the two different models (Baseline and Poisson).

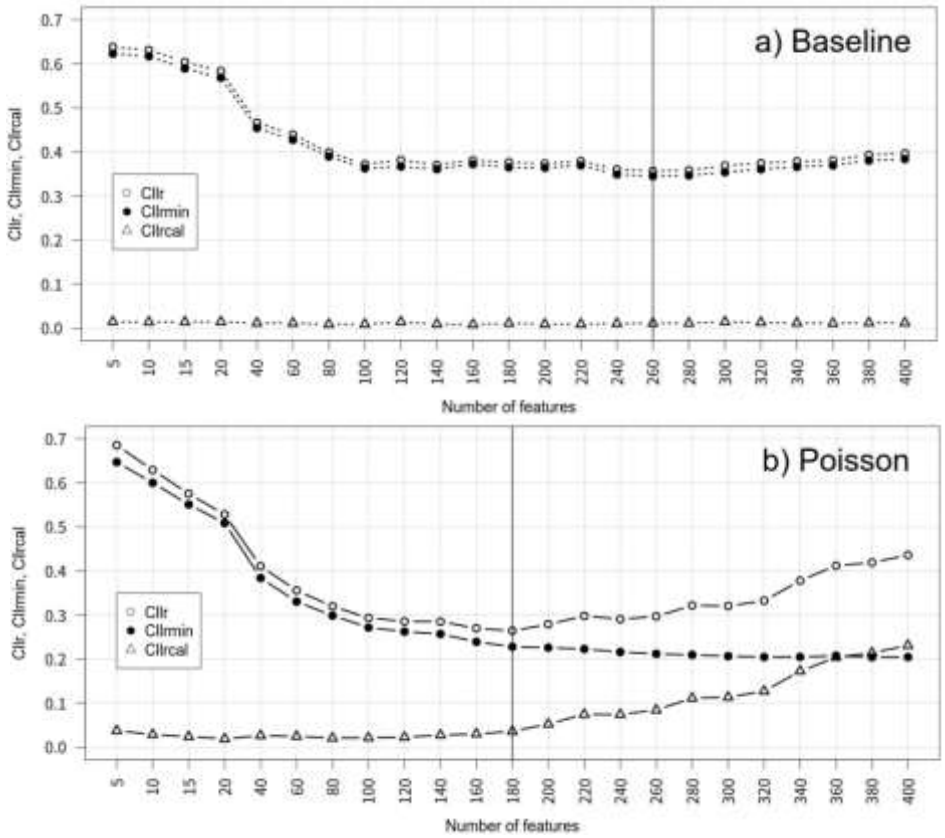


Figure 3: The C_{llr}, C_{llr}^{min} *and* C_{llr}^{cal} values of the LRs, with the *N* number of features indicated in the *y*-axis, are plotted separately for the Baseline (Panel a) and the Poisson (Panel b) models. The features are sorted according to word frequency. The vertical solid line indicates where the best C_{llr} value was obtained.

The calculation of C_{llr} is given in Equation 6) (Brümmer and du Preez, 2006).

$$C_{llr} = \frac{1}{2}\left(\begin{array}{l} \left[\frac{1}{N_{SA}}\sum_{i}^{N_{SA}} log_2\left(1+\frac{1}{LR_i}\right)\right] \\ +\left[\frac{1}{N_{DA}}\sum_{j}^{N_{DA}} log_2\left(1+LR_j\right)\right] \end{array} \right) \qquad 6)$$

In Equation 6), N_{SA} and N_{DA} are the number of SA and DA comparisons, and LR_i and LR_j are the LRs derived from the SA and DA comparisons, respectively. C_{llr} takes into account the magnitude of the LR values, and assigns them appropriate penalties. In C_{llr}, LRs that support the counter-factual hypotheses or, in other words, contrary-to-fact LRs (LR < 1 for SA comparisons and LR > 1 for DA comparisons) are heavily penalised and the magnitude of the penalty is proportional to how much the LRs deviate from unity. Optimum performance is achieved when C_{llr} = 0 and decreases as C_{llr} approaches and exceeds 1. Thus, the lower the C_{llr} value, the better the performance.

The C_{llr} measures the overall performance of a system in terms of validity based on a cost function in which there are two main components of loss: discrimination loss (C_{llr}^{min}) and calibration loss (C_{llr}^{cal}) (Brümmer and du Preez, 2006). The former is obtained after the application of the pooled-adjacent-violators (PAV) transformation – an optimal non-parametric calibration procedure. The latter is obtained by subtracting the former from the C_{llr}. In this study, besides C_{llr}, C_{llr}^{min} and C_{llr}^{cal} are also referred to.

The magnitude of the LRs derived from the comparisons are visually presented using Tippett plots. Details on how to read a Tippett plot are given in §5 when the plots are presented.

4 Results and Discussion

The C_{llr} values are plotted as a function of the number of features, separately for the Baseline model and the Poisson model in Figure 2. The number of the features is incremented by 5 from $N = 5$ to $N = 20$, and then by 20 from $N = 20$ to $N = 400$. For example, $N = 5$ means that the overall LRs were obtained by fusing the LRs derived with the five most-frequently occurring words for the feature-based method. Whereas the scores, which are to be converted to the LRs, were measured based on the vector of the five most-frequent words for the score-based method.

As can be observed from Figure 2, the performance of both models improves *en masse* as the N increases until a certain N, after which the performance remains relatively unchanged (or falls slightly). The Baseline model's performance stays relatively stable for a higher number of N, while the performance of the Poisson model begins to decline after 180 features. Due to the deterioration with a large number of feature numbers, although the Poisson model outperforms the Baseline model overall, the Baseline model does better with $N >$ 340.

The best performance, however, was observed for the Poisson with a lower number of features (C_{llr} = 0.26439; N = 180) relative to the Baseline model (C_{llr} = 0.35682; N = 260). The superior performance of the feature-based method (Poisson model) relative to the score-based method (Baseline model) conforms to the reports of previous studies on other types of evidence (Bolck et al., 2015; Morrison and Enzinger, 2018).

As described earlier, the Baseline and the Poisson models exhibit different performance characteristics in terms of the number of features required for optimal C_{llr} and the effect of increasing N. The performance of the Baseline model stays relatively unchanged with more features, while the performance of the Poisson model continuously declines with more features. In order to further investigate this performance difference, the C_{llr}, C_{llr}^{min} and C_{llr}^{cal} values are plotted separately for the two models in Figure 3.

For the Baseline model, it can be seen from Figure 3a that 1) the C_{llr}^{cal} values consistently remain close to 0, meaning the LRs are very well calibrated regardless of the number of features, and also that 2) the C_{llr}^{min} values display an almost identical trend as the C_{llr} values, meaning that like the C_{llr} values, the discriminability potential remains relatively constant even with an increase in the feature number after the best-performing point. In contrast, the three metrics plotted in Figure 3b reveal some notably different characteristics of the Poisson model. The C_{llr}^{cal} values stay low only until N = 140~160, after which the C_{llr}^{cal} values start increasing at a constant rate with an increase in the feature number; that is, the LRs become less well calibrated as N increases beyond 140~160 features. Unlike the calibration loss (and the Baseline model), the discriminability potential, quantified by C_{llr}^{min}, continues to improve at a small but constant rate, even after N = 180, where the best C_{llr} was observed. Thus, it is clear from Figure 3 that the deterioration of the Poisson model in performance after N = 180 is not due to a poor discrimination performance but due to a poor calibration performance. As explained in §3.5, logistic-regression fusion/calibration should theoretically yield well calibrated LRs. The poor calibration performance observed for the Poisson model for large feature numbers may be due to the interaction between the dimensions of the LRs to be fused and the amount of the training data for the fusion/calibration weights. This seems to be a typical example of the phenomenon known as the 'curse of dimensionality' (Bellman, 1961: p. 97), but further analysis is warranted. Nevertheless, it is clear that the use of a Poisson-based model, which theoretically better suits the distributional pattern of textual data and allows the rarity/typicality of evidence to be considered for LR estimation, can offer performance gains.

5 Feature Selection

For the Poisson model, LRs were first estimated separately for each of the 400 feature words. The resulting LRs were fused by gradually increasing the number of LRs included in the fusion set. LRs were arranged according to word frequency in the experiments reported in §4. Yet, the performance of a given feature (i.e. word) did not always correspond to the frequency of its occurrence. This is illustrated in Table 1, which lists the ten most frequently occurring words and the ten words with the highest discriminability (i.e. C_{llr}^{min}).

By word frequency		By C_{llr}^{min}	
Frequency	Words	Frequency	Words
1	'.'	3	','
2	'the'	1	'.'
3	','	41	'it's'
4	'and'	35	'!'
5	'i'	31	'-'
6	'a'	28	'('
7	'to'	27	')'
8	'it'	5	'i'
9	'of'	84	'i'm'
10	'is'	4	'and'

Table 1: Ten most-frequent (left) and lowest-C_{llr}^{min} (right) words

Thus, in this section, the words were first sorted according to their performance in terms of the C_{llr}^{min} values, and then the LRs were fused/calibrated based on the sorted words. The C_{llr} values of

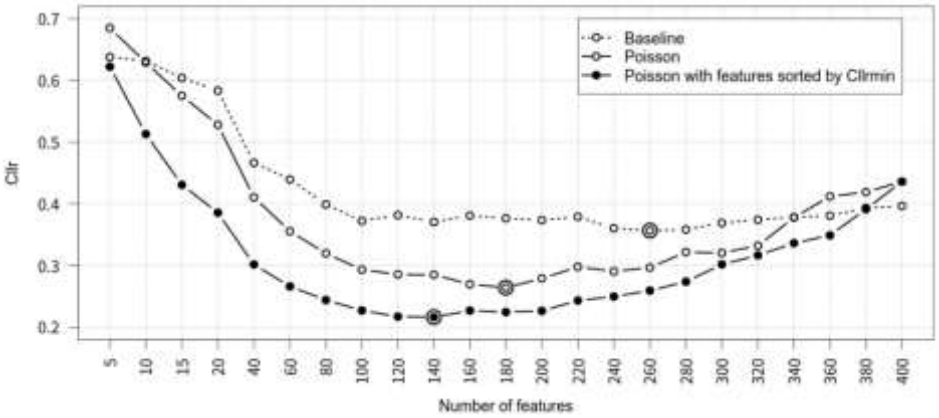


Figure 4: The C_{llr} values of the (fused) LRs with the N number of C_{llr}^{min}-sorted features indicated in the y-axis are plotted together with the results presented in Figure 2 for comparisons. The large circles indicate the best C_{llr} values for the models.

the experiments are plotted in Figure 4 including the results presented in Figure 2 for comparison.

It is clear from Figure 4 that selecting the features according to their C_{llr}^{min} values contributes to an improvement in performance for all numbers of features. As a result, the C_{llr} is lower (0.21664) with less features ($N = 140$) compared to the results with the unsorted features.

This feature selection approach was only possible because the LRs are estimated separately for the each of the 400 different words. This is possibly an advantage for the Poisson model.

The magnitude of the LRs with the best-performing settings are shown on Tippett plots, separately for the Baseline model, the original Poisson model, and the Poisson model with C_{llr}^{min}-sorted features in Figure 5. Tippet plots show the cumulative proportion of LRs from the SA comparisons (SALRs), which are plotted rising from the left, as well as of the LRs of the DA comparisons (DALRs), plotted rising from the right. For all Tippett plots, the cumulative proportion of trails is plotted on the y-axis against the $\log_{10}$ LRs on the x-axis. The intersection of the two curves is the equal error rate (EER) which indicates the operating point at which the miss and false alarm rates are equal.

As the low C_{llr}^{cal} values indicate, it can also be observed from Figure 5 that the LRs are very well calibrated. However, comparing Figure 5a and Figure 5bc we see that the magnitude of the LRs are weaker overall in the Baseline model compared to the two Poisson models; the Tippet lines are further from unity ($\log_{10}$ LR = 0) for the Poisson models than the Baseline models. Although the overall magnitude of LRs is greater for the Poisson models, unlike the Baseline model, they evince some very strong contrary-to-fact DALRs (which are indicated by arrows in Figure 5). This is a concern, and the reason for this needs to be further investigated.

6 Conclusions and Future Studies

A feature-based approach for estimating forensic LRs was implemented with a Poisson model for the first time in LR-based FTC. The results of the experiments showed that the feature-based FTC system outperforms the score-based FTC system with the Cosine distance. It has also been demonstrated that the performance of the feature-based system can be further improved by selecting the sets of LRs to be fused according to their C_{llr}^{min} values. It was observed that the discrimination loss in the feature-based FTC system reduces as the number of features increases, but becomes less well calibrated with a large number of features. It has been argued that this is a typical case of the 'curse of dimensionality' (Bellman, 1961: p. 97), but further investigation is required.

A simple one-level Poisson LR model shows good performance. However, it has been reported that word counts are often modelled poorly by standard parametric models such as the Binomial and Poisson models, and some alternatives have been proposed, such as the negative Binomial and the zero-inflated Poisson (Jansche, 2003; Pawitan, 2001). Alternatively, a two-level Poisson model might be implemented based if the prior distributions of λ is assumed (Aitken and Gold, 2013; Bolck and Stamouli, 2017). These alternatives should be tested to see if any improvements in performance are achievable.

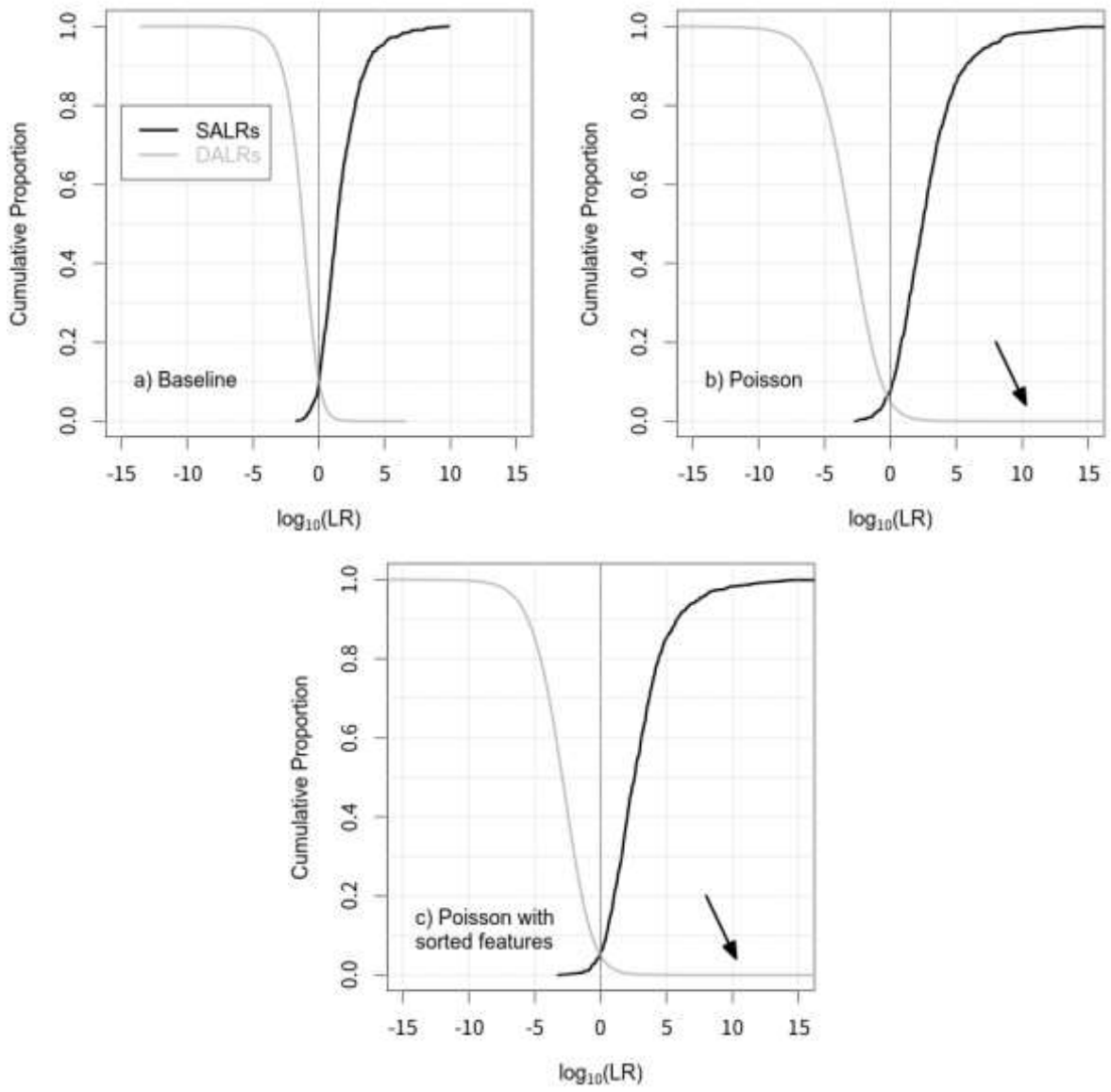


Figure 5: Tippett plots showing the magnitude of the derived LRs. Panel a) = Best-performing Baseline model; Panel b) = Best-performing original Poisson model; Panel c) = Best-performing Poisson model with sorted features according to their C_{llr}^{min} values. Note that some LRs extend beyond ±15 of the *y*-axis. Arrows indicate very strong contrary-to-fact DALRs.

The set of features tested in the current study is only one type of many potential authorship attribution features (according to Rudman (1997), over 1,000 different feature types have so far been proposed in the literature). While the purpose of the present study was to compare modelling approaches, rather than the relative performance of different feature types, an interesting future task would be to explore a richer feature set and the effect of different pre-processing techniques (e.g. stop word removal).

The LRs derived using the score-based method were well-calibrated, and therefore logistic-regression calibration was not necessary). This was not the case for LRs using the feature-based method where logistic-regression fusion/calibration was required. This procedure necessitates an extra set of data, namely a development database, and is another shortcoming of the feature-based method applied in this study

Acknowledgements

The authors thank the reviewers for their valuable comments. The first author's research is supported by an Australian Government Research Training Program Scholarship.

References

AbdulRazzaq, A. A. and Mustafa, T. K. (2014) Burrows-Delta method fitness for Arabic text authorship Stylometric detection. *International Journal of Computer Science and Mobile Computing 3*(6): 69-78.

Aitken, C. G. G. and Gold, E. (2013) Evidence evaluation for discrete data. *Forensic Science International 230*(1-3): 147-155. https://dx.doi.org/10.1016/j.forsciint.2013.02.042

Argamon, S. (2008) Interpreting Burrows's Delta: Geometric and probabilistic foundations. *Literary and Linguistic Computing 23*(2): 131-147. https://dx.doi.org/10.1093/llc/fqn003

Association of Forensic Science Providers. (2009) Standards for the formulation of evaluative forensic science expert opinion. *Science & Justice 49*(3): 161-164. https://doi.org/10.1016/j.scijus.2009.07.004

Bellman, R. E. (1961) *Adaptive Control Processes: A Guided Tour*. Princeton: Princeton University Press.

Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S. and Matsuo, A. (2018) quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software 3*(30): 774-776. https://doi.org/10.21105/joss.00774

Bolck, A., Ni, H. F. and Lopatka, M. (2015) Evaluating score- and feature-based likelihood ratio models for multivariate continuous data: Applied to forensic MDMA comparison. *Law, Probability and Risk 14*(3): 243-266. https://dx.doi.org/10.1093/lpr/mgv009

Bolck, A. and Stamouli, A. (2017) Likelihood ratios for categorical evidence; Comparison of LR models applied to gunshot residue data. *Law, Probability and Risk 16*(2-3): 71-90. https://dx.doi.org/10.1093/lpr/mgx005

Bolck, A., Weyermann, C., Dujourdy, L., Esseiva, P. and van den Berg, J. (2009) Different likelihood ratio approaches to evaluate the strength of evidence of MDMA tablet comparisons. *Forensic Science International 191*(1-3): 42-51. https://dx.doi.org/10.1016/j.forsciint.2009.06.006

Brümmer, N. and du Preez, J. (2006) Application-independent evaluation of speaker detection. *Computer Speech and Language 20*(2-3): 230-275. https://dx.doi.org/10.1016/j.csl.2005.08.001

Burrows, J. F. (2002) 'Delta': A measure of stylistic difference and a guide to likely authorship. *Literary and Linguistic Computing 17*(3): 267-287. https://dx.doi.org/10.1093/llc/17.3.267

Champod, C., Evett, I. W. and Kuchler, B. (2001) Earmarks as evidence: A critical review. *Journal of Forensic Sciences 46*(6): 1275-1284. http://dx.doi.org/10.1520/JFS15146J

Chen, X. H., Champod, C., Yang, X., Shi, S. P., Luo, Y. W., Wang, N., . . . Lu, Q. M. (2018) Assessment of signature handwriting evidence via score-based likelihood ratio based on comparative measurement of relevant dynamic features. *Forensic Science International 282*: 101-110. https://dx.doi.org/10.1016/j.forsciint.2017.11.022

Eder, M. (2015) Does size matter? Authorship attribution, small samples, big problem. *Digital Scholarship in the Humanities 30*(2): 167-182. https://doi.org/10.1093/llc/fqt066

Evert, S., Proisl, T., Jannidis, F., Reger, I., Pielström, S., Schöch, C. and Vitt, T. (2017) Understanding and explaining Delta measures for authorship attribution. *Digital Scholarship in the Humanities 32*(suppl_2): ii4-ii16. https://doi.org/10.1093/llc/fqx023

Evett, I. W. and Weir, B. S. (1998) *Interpreting DNA Evidence : Statistical Genetics for Forensic Scientists*. Sunderland, Mass.: Sinauer Associates.

Garton, N., Ommen, D., Niemi, J. and Carriquiry, A. (2020). Score-based likelihood ratios to evaluate forensic pattern evidence. *arXiv preprint arXiv:2002.09470*. Retrieved on July 20 2020 from https://arxiv.org/abs/2002.09470

HaCohen-Kerner, Y., Miller, D., Yigal, Y. and Shayovitz, E. (2018) Cross-domain authorship attribution: Author identification using char sequences, word unigrams, and POS-tags features. *Proceedings of Notebook for PAN at CLEF 2018*: 1-14.

Halvani, O., Winter, C. and Graner, L. (2017). Authorship verification based on compression-models. *arXiv preprint arXiv:1706.00516*. Retrieved on 25 June 2020 from http://arxiv.org/abs/1706.00516

Hepler, A. B., Saunders, C. P., Davis, L. J. and Buscaglia, J. (2012) Score-based likelihood ratios for handwriting evidence. *Forensic Science International 219*(1-3): 129-140. http://dx.doi.org/10.1016/j.forsciint.2011.12.009

Hoffmann, K. (1991) Statistical evaluation of the evidential value of human hairs possibly coming from multiple sources. *Journal of Forensic Sciences 36*(4): 1053-1058. https://dx.doi.org/10.1520/JFS13120J

Hoover, D. L. (2004a) Delta prime? *Literary and Linguistic Computing 19*(4): 477-495. https://dx.doi.org/10.1093/llc/19.4.477

Hoover, D. L. (2004b) Testing Burrows's Delta. *Literary and Linguistic Computing 19*(4): 453-475. https://dx.doi.org/10.1093/llc/19.4.453

Ishihara, S. (2014) A likelihood ratio-based evaluation of strength of authorship attribution evidence in SMS messages using N-grams. *International Journal of Speech Language and the Law 21*(1): 23-50. http://dx.doi.org/10.1558/ijsll.v21i1.23

Ishihara, S. (2017a) Strength of forensic text comparison evidence from stylometric features: A multivariate likelihood ratio-based analysis. *The International Journal of Speech, Language and the Law 24*(1): 67-98. https://doi.org/10.1558/ijsll.30305

Ishihara, S. (2017b) Strength of linguistic text evidence: A fused forensic text comparison system. *Forensic Science International 278*: 184-197. https://doi.org/10.1016/j.forsciint.2017.06.040

Jannidis, F., Pielström, S., Schöch, C. and Vitt, T. (2015) Improving Burrows' Delta. An empirical evaluation of text distance measures. *Proceedings of Digital Humanities 2015*: 1-10.

Jansche, M. (2003) Parametric models of linguistic count data. *Proceedings of Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*: 288-295.

Morrison, G. S. (2013) Tutorial on logistic-regression calibration and fusion: Converting a score to a likelihood ratio. *Australian Journal of Forensic Sciences 45*(2): 173-197. https://dx.doi.org/10.1080/00450618.2012.733025

Morrison, G. S. and Enzinger, E. (2018) Score based procedures for the calculation of forensic likelihood ratios - Scores should take account of both similarity and typicality. *Science & Justice 58*(1): 47-58. https://dx.doi.org/10.1016/j.scijus.2017.06.005

Morrison, G. S., Enzinger, E. and Zhang, C. (2018) Forensic speech science. In I. Freckelton and H. Selby (eds.), *Expert Evidence*. Sydney, Australia: Thomson Reuters.

Neumann, C., Champod, C., Puch-Solis, R., Egli, N., Anthonioz, A. and Bromage-Griffiths, A. (2007) Computation of likelihood ratios in fingerprint identification for configurations of any number of minutiae. *Journal of Forensic Sciences 52*(1): 54-64. https://dx.doi.org/10.1111/j.1556-4029.2006.00327.x

Omar, A. and Hamouda, W. (2020) The effectiveness of stemming in the stylometric authorship attribution in Arabic. *International Journal of Advanced Computer Science and Applications 11*(1): 116-121. https://dx.doi.org/10.14569/IJACSA.2020.0110114

Pawitan, Y. (2001) *In All Likelihood : Statistical Modelling and Inference Using Likelihood*. Oxford: Oxford University Press.

Rose, P. (2002) *Forensic Speaker Identification*. London: Taylor & Francis.

Rudman, J. (1997) The state of authorship attribution studies: Some problems and solutions. *Computers and the Humanities 31*(4): 351-365. https://dx.doi.org/10.1023/A:1001018624850

Rybicki, J. and Eder, M. (2011) Deeper Delta across genres and languages: Do we really need the most frequent words? *Literary and Linguistic Computing 26*(3): 315-321. https://dx.doi.org/0.1093/llc/fqr031

Smith, P. W. H. and Aldridge, W. (2011) Improving authorship attribution: Optimizing Burrows' Delta method. *Journal of Quantitative Linguistics 18*(1): 63-88. https://dx.doi.org/10.1080/09296174.2011.533591

Þorgeirsson, H. (2018) How similar are Heimskringla and Egils saga? An application of Burrows' delta to Icelandic texts. *European Journal of Scandinavian Studies 48*(1): 1-18. https://doi.org/10.1515/ejss-2018-0001

Vergeer, P., Bolck, A., Peschier, L. J. C., Berger, C. E. H. and Hendrikse, J. N. (2014) Likelihood ratio methods for forensic comparison of evaporated gasoline residues. *Science & Justice 54*(6): 401-411. https://dx.doi.org/10.1016/j.scijus.2014.04.008

Modelling Verbal Morphology in Nen

Saliha Muradoğlu[ΩΦ] **Nicholas Evans**[ΩΦ] **Ekaterina Vylomova**[μ]
[Ω]The Australian National University (ANU) [μ]The University of Melbourne
[Φ]ARC Centre of Excellence for the Dynamics of Language (CoEDL)
saliha.muradgolu@anu.edu.au, nicholas.evans@anu.edu.au,
ekaterina.vylomova@unimelb.edu.au

Abstract

Nen verbal morphology is remarkably complex; a transitive verb can take up to $1,740$ unique forms. The combined effect of having a large combinatoric space and a low-resource setting amplifies the need for NLP tools. Nen morphology utilises distributed exponence – a non-trivial means of mapping form to meaning. In this paper, we attempt to model Nen verbal morphology using state-of-the-art machine learning models for morphological reinflection. We explore and categorise the types of errors these systems generate. Our results show sensitivity to training data composition; different distributions of verb type yield different accuracies (patterning with E-complexity). We also demonstrate the types of patterns that can be inferred from the training data through the case study of syncretism.

1 Introduction

A long-standing research direction in NLP targets the development of robust language technology applicable across the wide variety of the world's languages. Unfortunately, the vast majority of machine learning models are being developed for a small fraction of nearly 7,000 languages in the world, such as English, German, French, or Chinese. With introduction of highly multilingual corpora such as UniversalDependencies (Nivre et al., 2016) and UniMorph (Sylak-Glassman et al., 2015; Kirov et al., 2018) the situation started to change. For instance, SIGMORPHON organized a number of shared tasks on morphological reinflection starting from 10 languages in 2016 (Cotterell et al., 2016) and up to 90 languages in 2020 (Vylomova et al., 2020). In 2020, languages were sampled from various typologically diverse families: Indo-European, Oto-Manguean, Tungusic, Turkic, Niger-Congo, Bantu, and others. Still, just one language, namely, Murrinh-patha, an Australian Aboriginal language (Mansfield, 2019), represented the whole linguistic variety of the Oceania region. In this paper, we aim at filling the gap by exploring Nen, a Papuan language spoken by approximately 400 people in Papua New Guinea. Nen is known for its rich verbal morphology, with a transitive verb inflecting for up to $1,740$ feature combinations. Distributed exponence, the phenomenon which gives rise to this large paradigm size, provides insight into modelling complex mappings between surface forms and feature bundles.

We conduct a series of experiments on morphological reinflection task recently introduced under the umbrella of SIGMORPHON (Cotterell et al., 2016, 2018). We train several state-of-the-art machine learning models for verbal inflection in Nen and provide an extensive error analysis. We investigate the relationship between the distribution of verb type (inflection classes) in the data and performance. Finally, we show that the system learns properties of the data that are not explicitly given, but may be inferred.

The rest of the paper is organized as follows: In Section 2, we give a brief overview of related work. Section 3 provides an overview of Nen verbal morphology, Section 4, details our methodology, and Section 5 presents our results. Finally, Section 6 concludes the paper.

2 Related Work

Muradoglu et al. (2020) is the only reported work on the computational modelling of the Nen language. Similar to this study, the main focus is on modelling Nen verbal morphology, but using finite-state architecture instead. The accuracy achieved by the FST system is 80.3% obtained across the corpus, with approximately 10% of the accuracy attributable to the modelling of prefixing verbs (the regularity of copula verbs boosts the accuracy from 70.5%). The accuracies reported are not directly

comparable with those presented here due to the different data splits, and increased amount of data.

In our error analysis, we follow the error taxonomy proposed by Gorman et al. (2019) upon a detailed analysis of typical errors produced by morphologically reinflection systems. A similar study was conducted for Tibetan (Di et al., 2019).

3 The Nen Language

Nen is a Papuan language of the Morehead-Maro (or Yam) family, located in the southern part of New Guinea (Evans, 2017). It is spoken in the village of Bimadbn in the Western Province of Papua New Guinea, by approximately 400 people, for which it is a primary language (Evans, 2015, 2020). Most inhabitants are multilingual, typically speaking several of the neighbouring languages.

The subject of this paper – verbs – are the most complicated word-class in Nen (Evans, 2015, 2019b). They are demarcated into three separate categories: prefixing, middle, and ambifixing verbs. The latter two are mostly regular in terms of morphophonological rules. In the remainder of this section, we elaborate on these characteristics, to give the reader enough background to follow the discussion in subsequent sections.

3.1 Verbal morphology

We begin our description from the maximal case – transitive ***ambifixing*** verbs. Examples of this verb type include *yis* 'to plant' and *waprs* 'to do' These verbs allow for full prefixing and suffixing possibilities. Evans (2016) provides the canonical paradigms for the undergoer prefixes, thematics and desinences. Suffix combinations are constructed by concatenating the corresponding thematic and the desinence. Between the undergoer prefix and verb stem is a directional prefix slot, available for all verb types. This slot is occupied by {*-n-*}[1] to convey a 'towards', {*-ng-*} for 'away' or left empty to convey a directionally neutral semantic.

Middle verbs such as *owabs* 'to speak' or *angs̄* 'to return', are also ambifixing, but the prefixal slot is restricted to {*n-*} (α–series), {*k-*} (β–series), {*g-*} (γ–series). These prefixes are person and number invariant, and mark the verb as being a dynamic monovalent verb. The prefix set is divided through the use of arbitrarily labels: α, β, and γ. These dummy indices do not carry specific semantic values until they are unified with other TAM (Tense, Aspect, and Mood) markings on the verb (Evans, 2015).

Prefixing verbs have separate closed paradigms, tailored to the subtype. Prefixing verbs are mostly distinguished through semantics; positional verbs such as *kmangr* 'to be lying down', the verb 'to own/have' *awans*, the verb 'to walk' *tan* and the copula verb *m* with its directional variants (be hither (i.e. come) or be thither (go)).

Inflectional prefixes for these verbs, mostly resemble the process with ambifixing verbs, yet the suffixes are limited. Of the 50 or so prefixing verbs, the vast majority are positional (Evans, 2020). An additional distinguishing feature of prefixing verbs, is the lack of infinitives. Both ambifixing and middle verbs form infinitives through suffixing *-s* to the verb stem. For the purposes of this study, we have listed the prefixing verb lemmas as the verb stem.

Methodologically, it is more convenient to segment a word as a classical bijective mapping between form to meaning. However, the Nen verbal system distributes information in a more complicated way. The prefixes (undergoer and future imperative) and suffixes (thematic and desinence) are not independent values. Nen verbal morphology is characterised by *distributed exponence (DE)*; "morphosyntactic feature values can only be determined after unification of multiple structural positions" (Carroll, 2016).

There are two consequences for morphological parsing:

a) Provisional unspecified values occur regularly, whether

   (i) These involve partial specification that will be filled in later in the word-parse, such as the left-edge prefix {*yaw-*} (1st person non-singular undergoer), which will only be made more precise in its number value (dual, or plural) when the thematic is encountered after the verb stem: thus *yaw-aka-t-an* 'I see them[2] (more them two)', where the 'non-dual' marker {*-ta-*} eliminates the dual (them two) but *yaw-akae-w-n* 'I see them

[1] We follow linguistic convention with '{}' denoting morphemes, and examples are italicised.

[2] Can also mean 'I see you (more than two)', resolved by combining with an appropriate free pronoun, *bm* 'you (absolutive)', but for present purposes we ignore this further complication.

(two)', where the 'dual thematic' {*-w-*} eliminates the plural (them more than two) reading.

(ii) These involve semantically-unspecified prefix series which only acquire meaning when they are combined with suffixes at the other end of the word: thus {*yaw-*}, in the above example, belongs to the α-series which, if it combines with the 'basic imperfective', will be given a (broadly) non-past reading, but when it combines with the 'past perfective' it will be given a past reading and when it combines with a 'projected imperative' it will be given a future meaning; a β-series form like {*taw-*}, by contrast, will have a 'yesterday past' interpretation when combining with the 'basic imperfective' suffixes but when combining with imperatives it will have a 'now/immediate command' meaning

b) More problematically, prefixes that normally have one reading (such as the yaw-example just discussed, which normally marks second/third person non-singular objects) sometimes have to be given a different meaning (e.g. large plural intransitive subjects) if further parsing to the right encounters a 'middle' rather than a 'transitive dynamic' stem (Evans 2017, 2019).

In principle that this means left-to-right morphological parsing is sometimes non-monotonic (particularly in the case of (b)), so that semantic values, as parsing proceeds, need to be sometimes held as provisionally unspecified, sometimes as partially specified, and sometimes as specified but subject to later override.

3.2 Distributed Exponence

One of the primary motivations for choosing Nen as a case study is the phenomenon that gives rise to this combinatorial power: distributed exponence. Essentially distributed exponence is a morphological phenomenon that gives rise to some types of non-monotonicity.

In linguistics, the notion of extended exponence was first introduced by Mathews (1974) and is now commonly referred to as multiple exponence (ME). Matthews defined ME as a category that would have exponents in two or more distinct positions. Distributed exponence is a kind of ME, which involves the use of more then one morphological segment to convey meaning. It requires all relevant morphs to yield a precise interpretation of the feature value in question (Carroll, 2016; Harris, 2017).

(1) n-ng-owan-t-e
M:α-VEN-set.off-ND:IPF.NP-IPF.NP.2|3SGA
'You/(s)he are/is setting off.'[3]

In the example above, no one marker marks the singular person. The information of the agent being singular is distributed across the thematic (dual/non-dual) and the desinence (single/dual/plural). If a non-dual thematic is present than the desinence cannot have dual features; the only options are singular or plural. Another morpheme present in this example is the prefix -ng- which marks the verb with the directional *thither*. The prefix n- marks this verb as a middle verb; it reduces the valency of the verb and yields information about the membership of the class α. Together with the prefix, thematic and desinence, the TAM feature can be obtained.

4 Methodology

4.1 Morphological reinflection task

Morphological inflection is a task of predicting a target word form from a corresponding word lemma and a set of morphosyntactic features (specifying the target slot, e.g. its part of speech (POS), tense, number, gender). For instance, a system is provided with a lemma "to sing" and a set of tags "Verb; Past" and needs to generate "sang". Morphological *reinflection* is a variation of the task when a lemma form is replaced with some other form and (optionally) its tags. The task has been traditionally solved with finite-state transducers, either hand-engineered (Koskenniemi, 1983; Kaplan and Kay, 1994) or trainable models that rely on both expert knowledge and data (Mohri, 1997; Eisner, 2002). In 2016 SIGMORPHON started a series of shared tasks on morphological reinflection, and neural models demonstrated superior performance when compared to finite-state or rule-based approaches, especially in high-resource languages (Cotterell et al., 2016; Vylomova et al., 2020).

[3] Example adapted from (Evans, 2020)

4.2 Data

The data used in this study comes from a Nen verb corpus (approximately $6,000$ verb samples representing $2,231$ unique inflected forms) created by Muradoğlu (2017). This dataset is a distilled subset from the approximately 8-hour natural speech corpus for the Nen language. As such it entails a frequency sorted list of all the verb forms occurring.

The training data is a set of triples comprising a lemma, morphosyntactic features, and an inflected form (i.e. we will only focus on morphological *inflection*).

Sampling Following the methodology in Cotterell et al. (2018) we split the data into training, development, and test sets. Training splits were created by sampling without replacement for three set sizes: all (*ALL*), medium (*MR*), and low (*LR*).

In virtue of coming from a natural corpus, the list of verb forms we use is Zipfian. This study does not distinguish between the feature bundles and only considered surface (inflected) forms. To facilitate the nature of our study, we uniformly distribute frequency across each syncretic cell.

For the *ALL* training set we start by sampling the first $1,931$ forms, in accordance with the Zipfian ranking across the corpus. In other words, we sample the $1,931$ most frequent verb forms. We randomly shuffle the remaining 300 forms into a 200 form test, and 100 form development (dev) sets. The test and dev sets remain the same through this experiment. Zipfian sampling is considered more realistic in this case, as it mimics the stimulus a language learner encounters. The dev and test set are randomly shuffled since supervised methods usually generalise from frequently encountered words.

For the *LR* and *MR* settings we take the first 100 and $1,000$ forms from the *ALL* training set, respectively. In addition, we create a high-resource (*HR*) set by supplementing the *ALL* set with synthetic forms, the final set contains $10,000$ forms. In order to generate synthetic samples, we use data hallucination technique proposed in Anastasopoulos and Neubig (2019). Note that the low-resource (*LR*) training set is a subset of the medium-resource (*MR*), which is supersetted by the *ALL* (and by extension the high-resource (*HR*) data set).

Finally, we contrast Zipfian sampling, when forms are sampled based on their frequency, to random sampling. Both sets (*LR* and *MR*) for the random sampling are created in a similar manner to Zipfian sampling, except frequency is not considered. Note that due to initial data size constraints, the *ALL* (and, therefore, *HR*) data sets for *both* the Zipfian and random sampling are the same. [4]

4.3 Experiments

In the current study we conducted three experiments to address our research questions.

4.3.1 Experiment 1: Testing across various data sizes and sampling methods

Research Question: How does training size and sampling method affect the models' performance, and what kind of errors are likely across these conditions?

We evaluate modelling accuracies across four different training sizes, which is further contrasted across sampling type. Our experimental setup mirrors those of the SIGMORPHON reinflection tasks (Cotterell et al., 2016, 2017, 2018; Vylomova et al., 2020): given an input lemma and a set of feature tags, models generate inflected forms. The final accuracy is computed as the percentage of matches between the gold and predicted forms.

4.3.2 Experiment 2: Testing compositionality of training data

Research Question: Does the composition of the training data affect the resultant accuracies, and, if so, how?

We test the effects of the verb type composition (i.e. how much of each verb type there is) in the training set. This study consists of seven (arising from all combinations of the three verb types) training data sets obtained through the sampling methods outlined above. We compare training sets of ambifixing verbs only, prefixing verbs only, middle verbs only, a two-way combination of each verb class: ambifixing and prefixing verbs, ambifixing and middle verbs, and prefixing and middle verbs and, finally an equal distribution of all three verb types, as listed in Table 4. Each set contains 386 forms (instances), stipulated by the amount of prefixing verbs available. The test and development set are 100 forms each, and is made up of 34 ambifixing, 33 middle and 33 prefixing verbs [5]

[4] Since the test and dev set are the same for both sampling methods, and are generated from the **remaining** 300 tokens (i.e. the least frequent items), it renders the random sampling of the *ALL* (and thus *HR*) the same.

[5] Uniform distribution is unlikely in natural language, in fact, Muradoğlu (2017) shows that the distribution is skewed

4.3.3 Experiment 3: Testing syncretism

Research Question: Do the models infer properties of the language which are not annotated in the data?

In Nen, the second and third person feature bundles often correspond to the same surface form across the available TAM categories (i.e. are syncretic). We test the likelihood of both models predicting the *unseen* second person singular for the past perfective TAM category as syncretic with the *seen* third-person singular variant. This is the one instance across the Nen verbal paradigm where this syncretism does not hold. In essence, we examine linguistic patterns that may be inferred from an annotated dataset.

The main focus here, is to categorise the type of prediction rather than the overall accuracy, as such training and development sets are identical to those generated for the *ALL* setting in the first experiment. The test set is comprised of 100 inflections of the past perfective second singular tags, most of these have been gathered from the Nen dictionary (Evans, 2019a).

4.4 Models

For our experiments, we will utilise two models that have shown superior performance in SIGMORPHON–CoNLL 2017 Shared Task on morphological reinflection in low- and medium-resource settings (Cotterell et al., 2017). Both of them are essentially neural sequence-to-sequence models implemented in Dynet (Neubig et al., 2017). In addition, we also compare the results with a simple non-neural baseline used in 2017–2018 tasks on morphological reinflection (Cotterell et al., 2017, 2018).

Hard Monotonic Attention (Aharoni and Goldberg, 2017) An external aligner (Sudoh et al., 2013) first produces transformation operations between an input (lemma) and a target (inflected form) character sequences. The alignment operations (steps) are then fed into a neural encoder–decoder model. The network, therefore, is trained to mimic the transformation steps, and at inference time it predicts the actions based on the input (lemma) sequence. Unlike soft attention models, this model attends to a single input state at each step and either writes a symbol to the output sequence or advances its pointer to the next state. Hard attention models demonstrate superior performance in languages that employ suffixing morphology with stem changes.

Neural Transition-based (Makarov and Clematide, 2018) The model is essentially derived from Aharoni and Goldberg (2017) by enriching it with explicit insertion, deletion or, alternatively, copy mechanisms. The copy mechanism led to significant accuracy gains in low-resource settings. Following Rastogi et al. (2016), the model can be seen as a neural parameterization of a weighted finite-state machine.

Non-neural Baseline (Cotterell et al., 2017, 2018) The non-neural system first aligns lemma and inflected form strings using Levenstein distance (Levenstein, 1966) and then extracts prefix- and suffix-based transformation rules.

4.5 Settings

The hyperparameters of the models are set to the values reported in the corresponding papers as per Table 1.

Hyperparameters	A&G	M&C
Input dim	100	100
Hidden dim	100	100
Epochs	100	50
Layer	2	1

Table 1: Hyperparameters for both A&G (2017) and M&C (2018) models.

5 Results

Table 2 shows the accuracies achieved for each system for each training set size and sampling type from Experiment 1. For all setups the M&C model performed best with random sampling (where applicable). As expected the high-resource setting performs best overall. The random sampling yields slightly higher accuracies than the Zipfian counterpart, this is likely due to the fact that prefixing verbs, particularly the copula and its 40 distinct forms occupy a majority of the top 100 positions in the Zipfian distribution. Thus when random sampling is utilized the training set includes more examples of ambifixing verbs.

to favour a higher number of ambifixing verbs in terms of the number of inflected forms.

	A&G 2017		M&C 2018		Non-Neural baseline (NNB)	
	Random	Zipf	Random	Zipf	Random	Zipf
HR	0.610		**0.650**		0.015	
ALL	0.390		**0.510**		0.010	
MR	0.295	0.285	**0.445**	0.420	0.000	0.000
LR	0.020	0.005	**0.080**	0.030	0.010	0.010

Table 2: Data set, model and sampling accuracies. ALL is a total of 1,931 verbs, HR is 10,000, MR is 1,000 and LR is 100 samples for the training set.

	ALL			HR			MR			LR		
	A&G	M&C	NNB	A&G	M&C	NNB	A&G	M&C	NNB	A&G	M&C	NNB
Allomorphy	56	55	190	54	46	144	61	77	188	17	162	190
Free Variation	30	24	0	14	15	11	13	24	0	0	2	0
Target	8	8	8	8	8	8	8	8	8	8	8	8
Stem	28	11	0	2	1	5	61	7*	2	174†	22	0
Total	122	98	198	78	70	168	143	116	198	199	194	198

Table 3: Absolute number of errors on the test set (200 instances) made by each system trained in ALL, HR, MR and LR setting.*contains 5 looping errors,† 17 looping errors.

5.1 Error Analysis

We analysed the errors produced in prediction following the taxonomy laid out by Gorman et al. (2019); Di et al. (2019).

We have taken a hierarchical approach to our error classification; whereby if more than one error is present, the category higher up is reported. For example, if a predicted form exhibits both target and allomorphy errors (error types are described in the following subsections), then only the target error is reported. The motivation for this lies in the nature of the error; free variation is technically not even an error. By contrast, misapplication of a morphophonological rule does indeed yield an incorrect form. Additionally, we have marked Target errors higher up as the system cannot be expected to correctly predict a form if the gold standard is incorrect. The hierarchy is as follows: Target>Stem>Allomorphy>Free Variation. 3 Table 3 summarises the types of errors across the different training sizes for each model. Overall, for both systems allomorphy errors remain relatively unimproved between the *ALL* and *HR* setting, but show a leap of reduction from the LR to MR conditions. Free variation errors are more prevalent in the *ALL* setting. This is probably a consequence of seeing more of the golden data and thus observing more of the systematic variations. This also explains why these errors reduce in number for the *HR* setting. The target errors are consistent across each experiment, as these are systematic issues with the gold data. Interestingly, stem errors reduce in the HR setting. This is despite the use of hallucinated data.

5.1.1 Allomorphy

This category consists of errors which are characterised by a misapplication of morphophonological rules, or feature category mappings. Frequent errors include the absence of vowel harmony or place assimilation rules, and incorrect mapping of feature bundles to surface forms. Most errors are of this category.

Vowel harmony. The Nen language exhibits vowel harmony. Consider the form *ynḡite* generated by one of the models, in a canonical sense the inflection is correct, but the presence of the high front vowel *i* requires the general *e* to harmonize to become *ynḡiti*.

Morphophonological Rules. When combining *r* final stems with *t* phonemes (which occurs in inflections via the non-dual thematics or certain desinences with ∅ thematics), the resultant sound is *n* (Evans, 2016). The M&C systems predicts that the stem *tar* inflected for the non-prehodiernal, first person actor and third person undergoer as *ytaretan*. Presumably, the break down is *y-tar-e-ta-n*. Interestingly, it inserts an *e* between the *r* and *t*, rather than concatenates the stem with the {-ta-n}

suffix. The correct form is *ytanan*.

Misapplication of category. These errors are rather straightforward: they are a misapplication of inflection rule and result in an incorrect cell of a paradigm. For example, *ynrenzan* is generated instead of *ynrenzng*. Technically, the form generated is correct, but it should correspond to the past perfective, first-person singular acting on dual actor suffix. Instead, it is mapped to the imperfective non-prehodiernal, third singular acting on dual actor suffix.

Future Imperatives. In all settings, across all systems tested, the future imperative was incorrectly predicted. Much like the β and γ counterparts, the system generated this TAM category by simply choosing an α prefix and suffixing {-ta}. Both A&G and M&C systems produce *yngita* instead of *yngangwita*. This formulation is correct for the β and γ series producing the imperfective imperative and mediated imperative, respectively. However, the future imperative has a special prefix which prefixes after the undergoer and directional prefix. It signifies the future imperative TAM category and marks the agent as either singular or non-singular.

Prefixing Verbs. Given the sparsity of examples for prefixing verbs and in particular their subtypes, a common occurrence across the data sizes is for the prefixing verb predictions to be inflected with the wrong features. For example, when the verb *m* 'to be' is inflected for the andative, 3PL+ undergoer and imperfective non-prehodiernal TAM the correct inflected form would be *yenewelmän*, instead the system gives *ynm* which it correctly identifies as a related form, but it does not have the correct inflectional features.

5.1.2 Free variation

Free variation errors occur when more than one acceptable inflected form exists; this is particularly true of the data set used in this study. The corpus used here has been distilled from a natural speech corpus, that has been transcribed. In addition to spelling variation - that arose as the orthographic decisions changed with ongoing documentation, the corpus also exhibits inter-speaker variation. An example includes: *yérniwi* as the predicted form and *yrniwi* as the gold standard. In Nen orthography, epenthetic vowels are not written in as their locations can be predicted (Evans and Miller, 2016). Older transcriptions wrote these vowels in with the *é*.

5.1.3 Target

These errors are characterised by incorrect feature tags in the gold standard data. One such example is as follows: the model predicts the form to be *nnganztat* and the gold standard is given as *ynganztat*, the feature tag, however, includes [M][6] and not [3SGU]. In such cases, based on the feature bundle, the system generated form is correct. This particular mismatch of middle and transitive verbs is the main source of this kind of error; it arises from the fact that a single verb may have a middle and transitive verb variant. This distinction can be difficult to decipher, and on some occasions, it can even be a result of speaker error.

5.1.4 Stem

This category denotes either a generated stem or a re-mapping of a seen but irrelevant stem, to the inflected form. These errors have linguistically viable morphemes attached, but we have not evaluated the accuracies of the mapping between feature and form for the morphemes.

Remapping

One such example is A&G model generating *ygmtandn* for the stem *sns*. It appears that the *gms* stem has been (incorrectly) inflected and mapped to the feature bundle of the *sns* stem. The correct inflection is *ysnendn*.

Generated Stem

The less frequent of the two are stems that have been randomly generated. For example with the stem given as *renzas*, the system generates: *ymryawem* in place of *yrenzawem*.

We have also encountered several looping errors such as: *ynawemaylmyylmyylmyylmyylmyylmyymayamawemyymamyamawemyymamyamawemyymamyamawemyymamyamawemyymamyamawemyylmyamyamawemyymamyamawemyymamyamawemyylmyylmyy* where the correct form is *ysnewem*.

5.2 Composition study

In Experiment 2, we tested the effects of training set composition; in other words, the informative nature of each verb type.

As mentioned above, the ambifixing verb class has the largest combinatorial space, reducing in size as we consider middle and prefixing verbs, respectively. Another way to consider this would be

[6] [M] marks the verb as middle and is present when one of the three middle prefixes is present.

	A&G	M&C	NNB
Ambifixing only	0.111	0.170	0.010
Middle only	0.121	0.210	0.111
Prefixing only	**0.212**	0.250	0.010
Ambi + Pre	0.111	0.190	0.010
Ambi + Mid	0.071	0.130	0.040
Mid + Pre	0.141	**0.290**	0.040
Ambi + Mid + Pre	0.061	0.200	0.040

Table 4: Data sets for each composition type, model and sampling accuracies. The training size for each is 386 forms (defined by the available prefixing verbs).

by providing comprehensive lists of the morphemes in a given language (such as Bickel and Nichols (2005); Shosted (2006)). Thus, the complexity of an inflectional system is measured by enumerating the number of inflectional categories and the range of available markers for their realisation (i.e. E-complexity). The bigger the number, the more complex the resulting system is.[7] With this in mind, we would expect that, given the same training size for each verb type, the ambifixing would perform the worst,[8] then the middle followed by the prefixing verbs. Our results, shown in Table 4, confirm this hypothesis.

More revealing than the overall accuracy for each set and model combination, is a decomposition of accuracy according to the verb class. Table 5 summarises the performance for each category according to verb class. Unsurprisingly, when the training set contains only one type of verb, it performs best for the type of verb seen in the training data.

From a linguist perspective, with principle parts from the middle verbs (mainly the suffixal system, recall that the middle verb takes a dummy prefix to reduce valency) and prefixing verbs (prefixal paradigm) we can construct the full paradigm available to ambifixing verbs. The results presented here show no such compositionality; instead, we see a simple correspondence to verb type observed.

As expected, we see the weak *leaking* or overlap between ambifixing and middle verbs, with very little transferability from prefixing to other verb types. It highlights the importance of tag choice; middle verbs have a [M] tag for the undergoer prefix, to mark the dummy prefix. If this tag were absent, would we see more transferability between ambifixing and middle verbs? Linguistically, no information would be lost as the absence of this tag still allows for the middle verbs to be clustered together.

5.3 Syncretism test

Experiment 3, entailed testing the systems with an unseen feature bundle and analysing the predicted forms, to gauge whether the models learnt syncretic behaviour.

As can be seen by the suffixal paradigm found in Evans (2016),[9] where both numbers are available, almost all the TAM categories exhibit syncretism across the second and third-person singular actor. The past perfective slot is the only case with distinct forms for the second and third singular person numbers. We are testing the prediction of an exception. The second singular is formed with {-nd-∅-} and the third person singular with the {-nd-a} suffix. We note the similarity between the second singular and dual forms, where the second dual is {-a-nd}. This becomes particularly pertinent when a vowel is inserted between consonants for ease of articulation but must also adhere to vowel harmony. In such cases, the second dual and second singular may appear the same.

Using the Aharoni and Goldberg (2017) architecture, the model incorrectly predicts 81 out of the 100 test forms as the third singular perfective category with the suffix {-nd-a} instead of {-nd–∅-}. Four forms predicted correctly (likely due to the similarity between the surface forms of the second person dual and singular tags) and the remaining fifteen distributed across second person dual and plural actor of the same TAM category, second/third singular for the imperfective non-prehodiernal TAM category, and several instances of nonce inflections such as {-ngt} or {-ngw}.

Similarly, the Makarov and Clematide (2018) system overwhelmingly predicts the unseen second singular form to be syncretic with the third singular (90 out of the 100 forms are predicted as such). Of the remaining ten instances three are correct, four are incorrectly modelled as the imperfective imperative (yet given the prefixing series is α, the future imperative prefix is absent) and one of

[7] Although more recent works have explored the issues with E-complexity (Ackerman and Malouf, 2013), we use it here as a guiding principle and acknowledge that further work is required to make a more nuanced statement.

[8] The combinatorial space for a transitive verb is $1,740$ cells (Muradoglu et al., 2020)

[9] Table 23.14 (pg 563) and Table 23.16 (pg 565)

	Ambifixing		Middle		Prefixing	
	AG	MC	AG	MC	AG	MC
Ambifixing only	11	15	2	0	0	2
Middle only	2	1	12	19	0	1
Prefixing only	0	0	0	0	21	24
Ambi + Pre	1	1	1	0	10	18
Ambi + Mid	1	4	6	8	0	1
Mid + Pre	0	3	3	10	11	16
Ambi + Mid + Pre	0	6	4	8	3	6

Table 5: Absolute number of correct predictions for each setup.

each: second/third imperfective non-prehodiernal, second/third neutral preterite or second dual past perfective.

From these results, it is clear that such systems not only observe patterns that are directly stipulated through annotation but also others that may be inferred from the data. It is important to note this behaviour, particularly in cases such as the one presented here as the verb corpus only entails two instances of the second singular past perfective.

6 Conclusion

Diversity representation of languages in NLP is vital to test the generalisations of models. We present the first-ever neural network-based analysis of Nen, the first representation of the Yam language family and to the best of our knowledge, of a Papuan language. Nen provides an interesting case study as it exhibits non-monotonic morphological mapping: distributed exponence.

We compare state-of-the-art models for morphological reinflection across various training sizes and two sampling methods: random and Zipfian. The results show no significant difference between sampling methods, and minor differences may be attributed to training set composition differences. In the Zipfian case, the prefixing verb types are over-represented as they are more frequent in natural speech. We provide extensive analysis of types of errors generated by each system and show that the most common error type is allomorphy errors; a misapplication of morphophonological rules, or feature category mappings. We introduce a new subcategory of error type: free variation, which is a consequence of the natural speech origins of the corpus.

We further explore composition effects by generating training sets with incremental distributions for the three verb classes noted. As expected, we found that the models trained with one class had higher prediction accuracy for that class. Across homogeneous compositions, the prefixing verb class performed the best. This is likely due to a smaller E-complexity – or more simply – a smaller combination of feature tags for which the system must learn mappings. Finally, we explore the likelihood of learning syncretic behaviour and using this as a predictor for an unseen feature bundle – the second singular past perfective. Overwhelmingly, the system incorrectly predicts syncretism with over 80% for the A&G system and 90% for the M&C system. These results highlight that these systems can infer patterns from the data sets provided. Although in our case the prediction of syncretism mirrors that of a human learner, there may be underlying, unwanted properties learnt from the data given, which calls for careful preparation of data and observation of output.

References

Farrell Ackerman and Robert Malouf. 2013. Morphological organization: The low conditional entropy conjecture. *Language*, pages 429–464.

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015, Vancouver, Canada.

Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the limits of low-resource morphological inflection. In *Proc. EMNLP*, Hong Kong.

Balthasar Bickel and Johanna Nichols. 2005. Inflectional synthesis of the verb. *The world atlas of language structures*, pages 94–97.

Matthew J. Carroll. 2016. *The Ngkolmpu Language*. Ph.D. thesis, The Australian National University.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27, Brussels. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared Task—Morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22, Berlin, Germany. Association for Computational Linguistics.

Qianji Di, Ekaterina Vylomova, and Tim Baldwin. 2019. Modelling Tibetan verbal morphology. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 35–40, Sydney, Australia. Australasian Language Technology Association.

Jason Eisner. 2002. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 1–8.

Nicholas Evans. 2015. Valency in Nen. In Andrej Malchukov, Martin Haspelmath, Bernard Comrie, and Iren Hartmann, editor, *Valency classes: A comparative handbook*, pages 1069–1116. Berlin: Mouton de Gruyter.

Nicholas Evans. 2016. Inflection in Nen. In Matthew Baerman, editor, *The Oxford Handbook of Inflection*, pages pages 543–575. Oxford University Press, USA.

Nicholas Evans. 2017. Quantification in nen. In *Handbook of Quantifiers in Natural Language: Volume II*, pages 571–607. Springer.

Nicholas Evans. 2019a. Nen dictionary. *Dictionaria*, pages 1–5005.

Nicholas Evans. 2019b. *Waiting for the Word: Distributed Deponency and the Semantic Interpretation of Number in the Nen Verb*, pages 100–123. Edinburgh University Press.

Nicholas Evans. 2020. Waiting for the word: distributed deponency and the semantic interpretation of number in the Nen verb. In Andrew Hippisley Matthew Baerman, Oliver Bond, editor, *Morphological perspectives*, pages 100–123. Edinburgh: Edinburgh University Press.

Nicholas Evans and Julia Colleen Miller. 2016. Nen. *Journal of the International Phonetic Association*, 46(3):331–349.

Kyle Gorman, Arya D. McCarthy, Ryan Cotterell, Ekaterina Vylomova, Miikka Silfverberg, and Magdalena Markowska. 2019. Weird inflects but OK: Making sense of morphological generation errors. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 140–151, Hong Kong, China. Association for Computational Linguistics.

Alice C Harris. 2017. *Multiple exponence*. Oxford University Press.

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sebastian J Mielke, Arya D McCarthy, Sandra Kübler, et al. 2018. Unimorph 2.0: Universal morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Kimmo Koskenniemi. 1983. *Two-level morphology: A general computational model for word-form recognition and production*, volume 11. University of Helsinki, Department of General Linguistics Helsinki, Finland.

Vladimir I Levenstein. 1966. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Peter Makarov and Simon Clematide. 2018. Neural transition-based string transduction for limited-resource setting in morphology. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 83–93, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

John Mansfield. 2019. *Murrinhpatha morphology and phonology*, volume 653. Walter de Gruyter GmbH & Co KG.

Peter H Mathews. 1974. *Morphology: an introduction to the theory of word-structure*. Cambridge, England: Cambridge University Press.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational linguistics*, 23(2):269–311.

Saliha Muradoglu, Nicholas Evans, and Hanna Suominen. 2020. To compress or not to compress? a finite-state approach to Nen verbal morphology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 207–213, Online. Association for Computational Linguistics.

Saliha Muradoğlu. 2017. *When is enough enough ? A corpus-based study of verb inflection in a morphologically rich language (Nen)*. Masters thesis, The Australian National University.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Pushpendre Rastogi, Ryan Cotterell, and Jason Eisner. 2016. Weighting finite-state transductions with neural context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 623–633, San Diego, California. Association for Computational Linguistics.

Ryan K Shosted. 2006. Correlating complexity: A typological approach. *Linguistic Typology*, 10(1):1–40.

Katsuhito Sudoh, Shinsuke Mori, and Masaaki Nagata. 2013. Noise-aware character alignment for bootstrapping statistical machine transliteration from bilingual corpora. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 204–209.

John Sylak-Glassman, Christo Kirov, David Yarowsky, and Roger Que. 2015. A language-independent feature schema for inflectional morphology. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 674–680, Beijing, China.

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrew Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, Christo Kirov, Garrett Nicolai, Adina Williams, Antonios Anastasopoulos, Hilaria Cruz, Eleanor Chodroff, Ryan Cotterell, Miikka Silfverberg, and Mans Hulden. 2020. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–39, Online. Association for Computational Linguistics.

An Automatic Vowel Space Generator for Language Learners' Pronunciation Acquisition and Correction

Xinyuan Chao[1], **Charbel El-Khaissi**[2], **Nicholas Kuo**[1], **Priscilla Kan John**[1], and **Hanna Suominen**[1, 3, 4]

[1]Research School of Computer Science, The Australian National University, Australia
[2]College of Arts and Social Sciences, The Australian National University, Australia
[3]Data61, Commonwealth Scientific and Industrial Research Organisation, Australia
[4]Department of Future Technologies, University of Turku, Finland

{*u6456596, charbel.el-khaissi, nicholas.kuo, priscilla.kanjohn, hanna.suominen*}*@anu.edu.au*

Abstract

Speech visualisations are known to help language learners to acquire correct pronunciation and promote a better study experience. We present a two-step approach based on two established techniques to display tongue tip movements of an acoustic speech signal on a vowel space plot. First, we use *Energy Entropy Ratio* to extract vowels; and then, we apply the *Linear Predictive Coding* root method to estimate Formant 1 and Formant 2. We invited and collected acoustic data from one *Modern Standard Arabic* (MSA) lecturer and four MSA students. Our proof of concept was able to reflect differences between the tongue tip movements in a native MSA speaker to those of a MSA language learner at a vocabulary level. This paper addresses principle methods for generating features that reflect bio-physiological features of speech and thus, facilitates an approach that can be generally adapted to languages other than MSA.

1 Introduction

Second language (L2) learners have difficulties in pronouncing words as well as native speakers (Burgess and Spencer, 2000) which can create inconveniences in social interactions (Derwing and Munro, 2005). Difficulty in providing pronunciation instructions by language teachers add extra challenges on L2 pronunciation training and corrections (Breitkreutz et al., 2001).

One solution to assist pronunciation acquisition is through the adoption of educational software applications (Levis, 2007). A well-designed language educational software can provide straightforward guidance to correct L2 pronunciation through multiple information sources. One instance of auxiliary systems is *Pronunciation Learning Aid* (PLA), which supports language students towards native-like pronunciation in a target language (Fudholi and Suominen, 2018). PLA achieves this via evaluating students' produced speech to reflect their pronunciation status. Another instance of auxiliary systems is visual cues, which serves as a friendly and accessible form of feedback to language students (Yoshida, 2018).

Through combining language lecturers' teaching with auxiliary systems, our aim is to assist students in both a classroom setting and in their individual practices. We present a prototype system that displays visual feedback on tongue movements to assist language learners to acquire correct pronunciation in the process of L2 studying. We have adopted a human-centred approach for the development of the system using a design-oriented perspective through applying a methodology that draws from *Design Science Research* (DSR) (Hevner et al., 2004) and *Design Thinking* (DT) (Plattner et al., 2009). Unlike machine learning methods, which train deep neural networks to predict articulatory movements (Yu et al., 2018), our proposed system uses vowel space plots based on bio-physiological features to help visualise tongue movements.

In this present work, we introduce a versatile prototype of our vowel space plot generator to address these challenges for students primarily learning MSA. Our design aims to allow L2 beginner learners to quickly visualise their status of pronunciation compared to those by their language teachers. We provide a reference vowel space plot adjacent to the students' own plots to reflect clear differences to support self-corrections. The envisioned applicability ranges from in-class activities to provide immediate and personalised suggestion to remote learning where in both cases glossary files are pre-uploaded by teachers or textbook publishers.

2 Related Work

Traditional acoustic plots, such as waveforms, spectrograms, and other feature plots are applied to vi-

sualise speech signals and can provide sufficient information to phoneticians, expert scientists, and engineers (Fouz-González, 2015). However, these methods fall short in providing straightforward suggestions for improving language students' pronunciation or otherwise lack an intuitive and user-friendly graphic user interface (Neri et al., 2002). A study proposed by Dibra et al. (2014) adopted the combination of waveform and highlighting syllables to visualise pronunciation in ESL studying shows using acoustic plots to support pronunciation acquisition is an implementable method.

Different from acoustic plots, another thinking of pronunciation visualisation was considered based on people's bio-physiological features. A pioneer study with this idea was introduced by Tye-Murray et al. (1993), in which they discussed the effect of increasing the amount of visible articulatory information, such as non-visible articulatory gestures, on speech comprehension. With the improvement of equipment, Ultrasound imaging, *Magnetic Resonance Imaging* (MRI), and *ElectroMagnetic Articulography* (EMA) can be alternative approaches to visualise the movement of articulators, and several study cases on pronunciation visualisation were implemented by Stone (2005), Narayanan et al. (2004), and Katz and Mehta (2015). However, these approaches are still difficult to be implemented in daily language studying since relevant equipment are often not available for in-class activities and self-learning, and generated images and videos are hard to be understood by ordinary learners.

Enlightened by imaging the movement of articulators, the idea of talking head, which is using 3D mesh model to display of both the appearance articulators and internal articulators, was introduced. Some of the fundamental works of talking head were completed by Odisio et al. (2004), and Serrurier and Badin (2008). With the techniques of articulatory movement prediction, such as *Gaussian Mixture Model* (GMM) (Toda et al., 2008), *Hidden Markov model* (HMM) (Ling et al., 2010), and popular deep learning approach (Yu et al., 2019). Although talking head is developing swiftly, the research about performance of talking head for pronunciation training is still insufficient.

The place and manner of articulation are well established variables in the study of speech production and perception (e.g. Badin et al., 2010). Early research has already realised the potential of using vowel space plots to achieve pronunciation visualisation, such as the studies by Paganus et al. (2006) and Iribe et al. (2012). These studies indicate that for language learners, vowel space plots are easy-to-understand, straightforward, and provide the necessary information for understanding their own tongue placement and movement. Therefore, vowel space plots are considered a useful tool for language learners to practice and correct their pronunciation relative to other pronunciation correction tools, such as ultrasound visual feedback or more traditional pedadogical methods like explicit correction and repetition.

3 The Proposed Approach

To visualise the tongue movement based on students' pronunciation practice, our proposed system needs to receive students' pronunciation audio signal as its input. After the process of vowel detection, vowel extraction, and formant estimation, the system can automatically generate the corresponding vowel space plot as its output. In this section, we will introduce how engineering and linguistics insights inspired our proposed method, and the details of audio signal processing procedures.

3.1 Design Methodology

To find a reliable solution for language students on the challenges about pronunciation acquisition, we adopted a design-based approach and implemented a human-centred approach by using the Design Thinking framework (Plattner et al., 2009) to find the students' needs in terms of pronunciation practice and transform these into requirements. In the Empathy and Define phases of DT, we defined our research question as "Finding an implementable and friendly approach for language learners to help them practice their pronunciation". After this, we participated in an MSA tutorial and observed students' behaviours during the process of pronunciation acquisition. Finally, we generated an online questionnaire for students which asks their in-class pronunciation training experience and their study preferences. The details of this survey were introduced in the thesis by Chao (2019).

Based on the observation of MSA tutorial, we found that students feel comfortable to interact with other people (lecturer or classmates) during pronunciation process. One advantage for interaction is other people can provide feedback on students' pronunciation. Another finding from observation

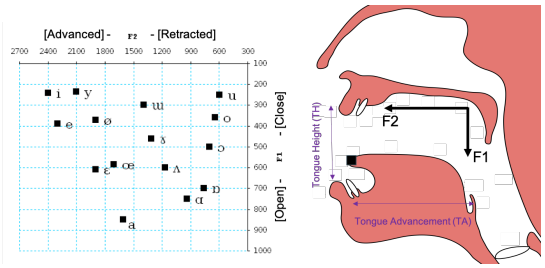
is the process of pronunciation acquisition can be seen as a process of imitation. Students need a gold-standard, such as teachers' pronunciation, as a reference to acquire new pronunciation and correct mispronunciation. The survey gives us some insights into students preferences about pronunciation study pattern. One of the most important insight is that students are interested in multi-source feedback of pronunciation training. For ordinary pronunciation, training students can only receive auditory information of pronunciation. Therefore, if a straightforward and easy-understanding visual feedback can be adopted in our proposed method, students will have a better experience and higher efficiency on pronunciation training.

The DT Empathy and Define phases gave us the insight that an ideal auxiliary pronunciation system should interact with learners, provide gold-standard pronunciation reference, and display reliable visual feedback to learners. The insight gained led to ideation discussions leading to the selection of vowel space plots as visualisation tool. We augmented the use of DT with the DSR approach, in the manner of John et al. (2020)'s study, to guide the development of our the artefact generated from our insights. Using the DSR method introduced by Peffers et al. (2007), we (1) identified our research question based on a research project which is about assisting new language learner on pronunciation acquisition with potential educational softwares, (2) defined our solution according to our observation and survey, (3) designed and developed our prototype of vowel space plot generator, (4) demonstrated our prototype to MSA lecturers and students, (5) and evaluated the prototype's performance. The DT and DSR process underpin all our methods.

3.2 Vowel Space Plot

Our proposed prototype uses vowel space plots as a tool to visualise the acoustic input. This visualisation then forms the basis for subsequent feedback on pronunciation features.

A vowel space plot is generated by plotting vowel formant values on a graph that approximates the human vocal tract (Figures 1(a) and 1(b)). F1 and F2 vowel formant values correlate with the position of the tongue during articulation (Lieberman and Blumstein, 1988). Specifically, F1 is associated with the height of the tongue body (tongue height) and plotted along the vertical axis, while its



(a) An example of vowel space (b) Vowel space plot and plot which shows the location oral cavity – the Formant- of different vowels in the vowel Articulation Correlation space

Figure 1: Vowel space plot and oral cavity

F2 counterpart is associated with tongue placement in the oral cavity (tongue advancement) and plotted along the horizontal axis.

The correlation between formant values and the tongue's height and placement is referred to as the formant-articulation relationship (Lee et al., 2015). These F1-F2 formant values can be rendered as x-y coordinates on a 2D plot to visualise the relative height and placement of the tongue in the oral cavity during articulation. When visualised alongside the tongue position of a native speaker's pronunciation, users can then see the position of their tongue relative to a standard reference or benchmark of their choice, such as an L2 teacher or native speaker. This visualisation supports pronunciation feedback and correction as users could then rectify the placement and/or height of their tongue during articulation to more closely align with its position in an equivalent native-like pronunciation.

3.3 Vowel Detection and Perception

To extract vowels from input speech signal, first, we calculate relevant energy criteria and find speech segments. Once speech segments were confirmed, we then use defined thresholds and detect vowels from these speech segments. This section will introduce the energy criteria and the thresholds we adopted in our practice.

Before detecting vowels in a speech signal, detrending and speech-background discrimination are two necessary steps of pre-processing. These steps ensure that only the correct speech information from the original signal is extracted, while other possible noise is ignored. In this way, the prototype minimises the possibility of including irrelevant signals during the feature extraction process.

Our prototype adopted the spectral subtraction

algorithm to achieve speech-background discrimination, as first introduced by Boll (1979). And the detrending can be achieved by the classic least squares method.

Our approach used *Energy Entropy Ratio* (EER), which is a calculated feature from input signal, as the criteria to find vowels from input speech signal. The EER can be calculated as following steps.

The *spectral entropy* (SE) of a signal describes its spectral power distribution (Shen et al., 1998). SE treats the signal's normalised power distribution within the frequency domain as a probability distribution and calculates its Shannon entropy. To demonstrate the probability distribution of a signal, let a sampled time-domain speech signal be $x(n)$, where the ith frame of $x(n)$ is $x_i(k)$ and the mth of the power spectrum $Y_i(m)$ is the *Discrete Fourier Transformation* (DFT) of $x_i(k)$. If N is the length of *Fast Fourier Transformation* (FFT), the probability distribution $P_i(m)$ of the signal can be then expressed as

$$p_i(m) = \frac{Y_i(m)}{\sum_{l=0}^{N/2} Y_i(l)}. \quad (1)$$

The definition of short-time spectral entropy for each frame of the signal can be further shown as

$$H_i = -\sum_{k=0}^{N/2} p_i(k) \log p_i(k). \quad (2)$$

The spectral entropy reflects the disorder or randomness of a signal. The distribution of normalised spectral probability for noise is even, which makes the spectral entropy value of noise great. Due to the presence of formants in the spectrum of signals in human speech, the distribution of normalised spectral probability is uneven, which makes the spectral entropy value small. This phenomenon can be used with speech-background discrimination to find out endpoints of speech segments.

In its practical application, SE is robust under the influence of noise. But spectral entropy cannot be applied for signals with a low *signal-to-noise ratio* (SNR) because when SNR decreases, the time-domain plot of spectral entropy will keep the original shape, but with a smaller amplitude. This makes SE insensitive to distinguishing speech segments from background noise. To provide a more reliable method of detecting the beginning and end of speech intervals, we introduce

$$EER_i = \sqrt{1 + |E_i / H_i|}, \quad (3)$$

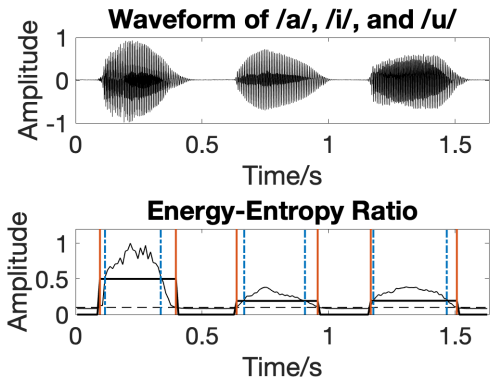


Figure 2: Vowel detection and segmentation

where E_i is the energy of the i^{th} frame of a speech signal, and H_i is the corresponding SE. Speech segments will have larger energy and smaller SE than silent segments. A division of these two short-term factors makes the difference between speech segments and silent segments more obvious.

The first threshold T_1 was implemented as the criterion to judge if the segment contains speech or not. The value of T_1 can be adjusted, and in our case we chose $T_1 = 0.1$ which performs well. Thus, segments with an energy entropy ratio larger than T_1 were classified as speech segments.

In each speech segment that is extracted, the maximum energy entropy ratio, E_{max}, and scale factor r_2, were used to set another threshold T_2 for detecting vowel segments:

$$T_2 = r_2 E_{max}. \quad (4)$$

Since different speech segments may have a different threshold T_2, segments with an energy entropy ratio larger than T_2 were used to detect vowels.

In an example visualisation of vowel detection and segmentation (Figure 2), three vowel phonemes — /a/, /i/, and /u/ — are contained in the speech signal. The black dashed horizontal lines show the threshold value $T_1 = 0.1$ for speech segment detection, while the solid orange lines show the detected speech segments within the speech signal. Similarly, the black vertical lines in bold indicate a dynamic threshold value T_2 for vowel detection across different speech segments, while the blue dashed lines display the vowel segments.

3.4 Formant Estimation

Formant value estimation is the next task after the detection of vowel segments from input speech signals. Our prototype adopted the *Linear Predictive*

Coding (LPC) root method to estimate the F1 and F2 formant values for vowels.

A common pre-processing step for linear predictive coding is pre-emphasis (highpass) filtering. We apply a straightforward first-order highpass filter to complete this task.

A simplified speech production model, which we adopted in our work is represented in Figure 3 following Rabiner and Schafer (2010). As shown in Figure 3, $s[n]$ is the output of the speech production system, $u[n]$ is the excitation from the throat, G is a gain parameter and $H(z)$ is a vocal tract system function. Let us consider the transfer function of $H(z)$ as an *Auto-Regression* (AR) model

$$H(z) = \frac{G}{A(z)} = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \quad (5)$$

where $A(z)$ is the prediction error filter, which is used in the LPC root method below.

The polynomial coefficient decomposition of prediction error filter $A(z)$ can be used to estimate the centre of formants and their bandwidth. This method is known as the LPC root method, which was first introduced by Snell and Milinazzo (1993). Notably, the roots of $A(z)$ are mostly complex conjugate paired roots.

Let $z_i = r_i e^{j\theta_i}$ be any value of a complex root of $A(z)$, where its conjugate $z_i^* = r_i e^{-j\theta_i}$ is one of the roots of $A(z)$. Further, if F_i is the formant frequency corresponding to z_i, and B_i is the bandwidth at 3dB, then we have the relationships $2\pi T F_i = \theta_i$ and $e^{-B_i \pi T} = r_i$, where T is sampling period. Their solutions are $F_i = \theta_i/(2\pi T)$ and $B_i = -\ln r_i/\pi T$.

Since the order p of prediction error filter is set in advance, the pair number of complex conjugate roots will be up to $p/2$. This makes it straightforward to find which pole belongs to which formant, since extra poles with a bandwidth larger than a formant's bandwidth may be conveniently excluded.

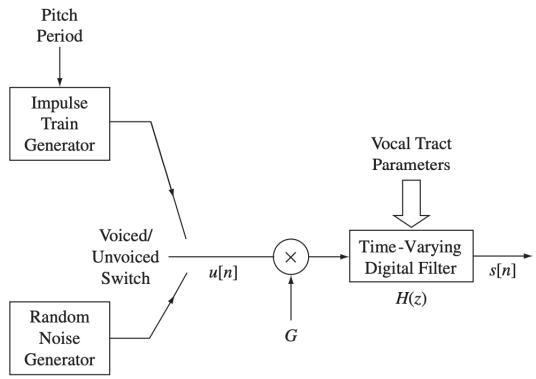


Figure 3: A simplified model of speech production

4 Preliminary Evaluation Experiment

We conducted two experiments to evaluate the performance of our prototype. First, we invited a native Arabic speaker who is a *Modern Standard Arabic* (MSA) lecturer at The Australian National University (ANU) to provide a glossary of MSA lexicon and their corresponding utterances. These utterances constituted the gold-standard or target pronunciation for users. Then, we invited four MSA language students to use our prototype by pronouncing four MSA words. For each lexical item pronounced, the articulation was visualised on a vowel space plot so users can compare their pronunciation alongside the native-like, target pronunciation of their lecturer. Following this visual comparison, users were prompted to pronounce the same word again.

In the experiments, we want to verify the feasibility and accessibility of our prototype. The feasibility of our prototype was determined by whether the interpretation of the comparison plots in the first instance supported improved pronunciation of the same word in subsequent iterations. And the accessibility refers to whether our prototype can provide implementable and correct feedback for learners to visualise their pronunciation.

Ethical Approval (2018/520) was obtained from the Human Research Ethics Committee of The Australian National University. Each study participant provided written informed consent.

4.1 Feasibility Test

The functionality of the prototype, including speech detection, vowel segmentation and plot generation, was first verified by using a series of acoustic signals as input to observe the accuracy of the output vowel space plot. The MSA lecturer's pronunciation of MSA lexicon was used here to test the veracity of the prototype output. The MSA dataset comprised of ten lexical items[1] and their corresponding pronunciation, henceforth referred to as the "standard reference" (see Table 1).

For each vocabulary item and corresponding audio input, we observed the vowel space plot gen-

[1]Refer to MSA Vocabulary Selection (Section 8) on our selection criteria of this list.

Vocabulary	MSA	Transliteration	Vowels
clock	ساعة	/sāʿa/	2
eggs	بيض	/bayḍ/	1
mosque	جامِع	/jāmiʿ/	2
phone	هاتِف	/hātif/	2
shark	قِرْش	/qirš/	1
soap	صابون	/ṣābūn/	2
spring	ربيع	/rabīʿ/	2
street	شارِع	/šāriʿ/	2
student(male)	طالِب	/ṭālib/	2
student(female)	طالبة	/ṭāliba/	3
watermelon	بَطّيخ	/bāṭṭīḵ/	2

Table 1: Ten reference vocabularies

Vocabulary	MSA	Transliteration	Vowels
shark	قِرْش	/qirš/	1
soap	صابون	/ṣābūn/	2
student(male)	طالِب	/ṭālib/	2
student(female)	طالبة	/ṭāliba/	3

Table 2: The student test data of four MSA words

erated by our prototype. The accuracy and accessibility of our prototype's speech and vowel detection functionality was determined by its ability to correctly visualise tongue positioning for each vowel in a word. This was determined based on a comparison with statistical averages of formant values for the same vowel. We use a Sony Xperia Z5 mobile phone to collect the utterance of glossary from the MSA lecturer. The utterances were recorded as individual mp3 files which can be used as input of our prototype. Each mp3 file contains one MSA vocabulary in the glossary. These mp3 files were recorded in the lecturer's office to reduce background noise.

4.2 Accessibility Test

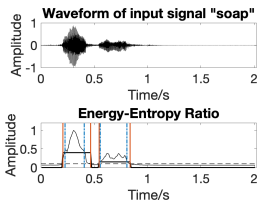
The verification of our prototype's functionality alone is insufficient to prove that the prototype can assist in providing valuable corrective feedback to users. Therefore, we invited two male students and two female students who were enrolled in a beginner MSA course (ARAB1003) at ANU to voluntarily participate in our accessibility test The success of our prototype's feedback function was determined by whether the language learners can interpret their pronunciation on a vowel space plot against the standard reference in order to produce a more native-like pronunciation for the same word.

Volunteers were aged between 19 and 22 and had completed an introductory MSA course (ARAB1002), which meant they had basic knowledge of MSA and were familiar with its alphabet and phonetic inventory. Four lexical items from the glossary in the standard reference were selected as test items which shown in Table 2 for the volunteers to pronounce. Volunteers pronounced each of the four vocabulary items independently, which were recorded respectively as audio files. These files were processed by our prototype and the corresponding vowel space plots were generated to visualise their pronunciation for each word. Then, their vowel space plots were compared to the corresponding vowel space plot of the standard reference. Participants were advised to use this comparison plot as the basis for their pronunciation feedback prior to repeating the pronunciation of the word. Then, participants pronounced the word a second time and the generated plot was once again compared to the standard reference. This time, the comparison assessed whether the participant's articulation of the vowel was more closely aligned to the standard reference compared to the first pronunciation. In other words, the second iteration of pronunciation allowed for an assessment of whether our prototype provided valuable visualisation information to participants, and whether it helped them immediately correct and improve their pronunciation relative to the standard reference.

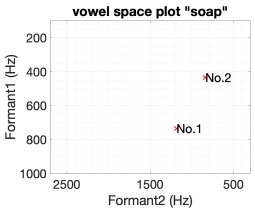
We participated in one of the MSA course tutorials and were keen to see the quality of acoustic data, which were collected from a noisy circumstance, like a classroom. The collecting device was a MacBook Pro 2017. We wrote a Matlab recorder function with GUI to collect the utterance provided by volunteers who were from this tutorial. The utterance were collected as individual wav files and each file contained one word from volunteers.

5 Results and Discussion

We used collected speech signals to test the feasibility and accessibility of our prototype. To test the feasibility, we fed the standard references to our prototype and verify whether the output vowel space plot can reflect the correct tongue motion of the corresponding word. As for accessibility, we used the student test data and generated the vowel space plot, and then found corresponding words from a standard reference and compare these two vowel space plots. An ideal result is the student test



(a) Vowel segmentation of standard reference "soap"



(b) Vowel space plot of standard reference "soap" with /ā/ and /ū/ two vowels

Figure 4: The waveform, energy-entropy ratio, and vowel space plot for standard reference word "soap" (provided by a MSA teacher)

data can reflect the student's tongue motion, and the student can find how to improve the pronunciation by compare these two vowel space plots. With the vowel space plots of the same words from student test data and standard reference, we compared the corresponding plots to see if the corresponding plots and if the vowel space plots can provide useful feedback on pronunciation correction. In this paper, we display the MSA word "soap" (صابون , /ṣābūn/) as an example of our results.

5.1 Feasibility

To test the feasibility of our prototype, we picked one vocabulary item (the word "soap") from standard reference and verify whether the output vowel space plot can reflect the tongue motion. The waveform, energy-entropy ratio, and vowel space plot for standard reference word "soap" (Figure 4).

From Figure 4(a), we found two voice segments between solid orange lines that were recognised from the input speech signal, and the two voice segments, which contained one vowel between dash blue lines for each. In Figure 4(b), the two vowels of /ā/ and /ū/ were mapped in the vowel space. This vowel space plot was made available to the users so they can get familiar with their tongue position in the oral cavity and use this visual feedback towards pronouncing the word "soap" correctly (Figure 5).

5.2 Accessibility

To test the accessibility of our prototype, we compared the vowel space plot of standard reference and the vowel space plot of student test data. We continue to use the word "soap" here as an example. Figures below show the results of MSA vocabulary "soap" pronounced by the four anonymous students. Students will see two vowel space plot from the

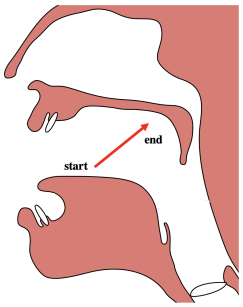


Figure 5: The tongue motion for the MSA word "soap"

prototype: one shows the standard reference, and another reflects their own pronunciation.

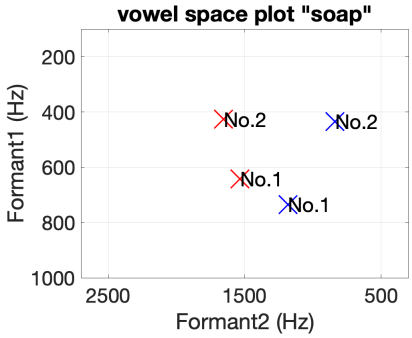
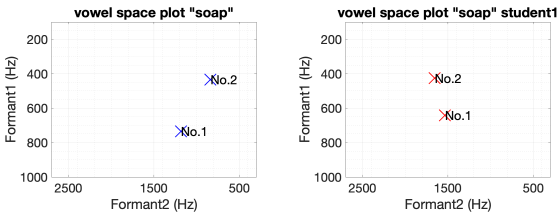


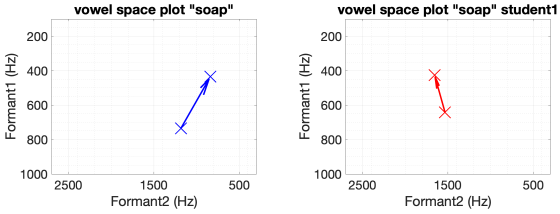
Figure 6: The tongue movement (reference and student1's practice) for the MSA word "soap"



(a) Standard reference "soap"

(b) Vowel space plot of user input-1 "soap" with /ā/, /ū/

Figure 7: The vowel space plot from standard reference and student1

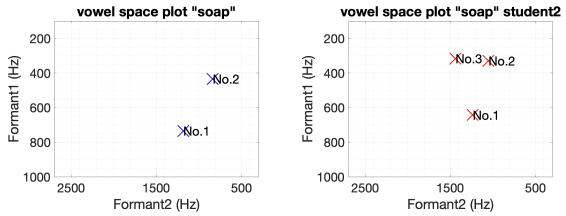


(a) Standard reference "soap" with arrow

(b) Vowel space plot of user input-1 "soap" with arrow

Figure 8: The vowel space plot from standard reference and student1 with arrow

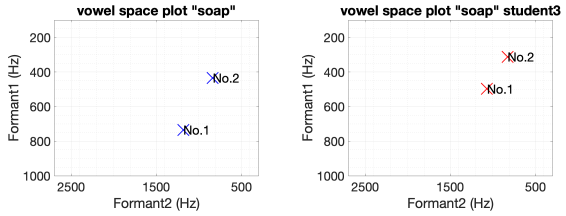
Figure 6 shows the overlay vowel space plot of standard reference (blue crosses) and student1's pronunciation practice (red crosses). Since the key information from vowel space plot is the trend of tongue movement, it is not necessary to compare the standard reference and students' pronunciation on the same vowel space plot. From Figure 7, student1's tongue should be drawn back instead of moving it to the front of the oral cavity. The vertical down-up movement of the tongue was correct. Figure 8 shows the tongue movement with an arrow. This is more readable and friendly for students to help them perceive their tongue movement.



(a) Standard reference "soap" (b) Vowel space plot of user input-2 "soap" with /ā/, /ū/ following wrong trajectory

Figure 9: The vowel space plot of standard reference and student2

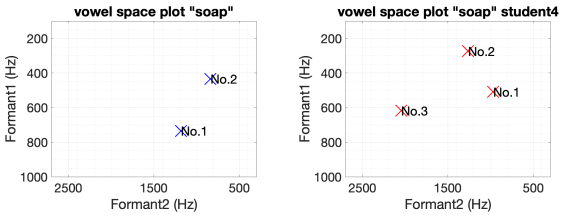
Student2, on the other hand, should focus on the pronunciation of the second vowel /ū/. According to Figure 9, we can see that the pronunciation of "soap" pronounced by student2 had the correct tongue motion trajectory when compared with the standard reference of Figure 1. This student's vertical down-up movement of the tongue was correct. A small defect for this practice was that there existed an unexpected vowel for the end of this pronunciation practice. For further practice, the advice for student1 targeted pronouncing a clean and neat end of the word "soap".



(a) Standard reference "soap" (b) Vowel space plot of user input-3 "soap" with /ā/, /ū/

Figure 10: The vowel space plots of standard reference and Student3

Student3, in turn, had the correct tongue motion, and the pronunciation was good as well. However, the starting point of the first vowel /ā/ was somewhat higher than its standard reference. Hence, our suggestion for Student3 was to lower the starting position of the word "soap".



(a) Standard reference "soap" (b) Vowel space plot of user input-4 "soap"

Figure 11: The waveform, energy-entropy ratio, and vowel space plot of Student4

Finally, student4 and student1 made similar mispronunciation: student4 should draw the tongue back instead of moving it forward while pronouncing the second vowel /ū/. Besides this mistake, another interesting point worthy of notice was that another unexpected vowel occurred by the end of this speech signal. According to waveform analysis, this vowel was not pronounced by student4 but originated from the background noise due to the data collection during an in-class activity. This meant that the sudden noise from background can still influence the analysis result although our prototype already applied its denoising algorithm to this speech signal. Hence, we made a suggestion to try to adopt a more effective denoising function as the future development of the system to satisfy the requirements from students to practice their pronunciation anywhere, including noisy settings.

6 Conclusion

This paper presented the initial proof of concept that used vowel space plots to enhance language learning in second languages. The idea of our prototype was based on our early stage DSR process and MSA language student survey (Chao, 2019). Our prototype was designed to generate clear visual feedback from speech input, and it was tested to assist the pronunciation of L2 MSA beginners.

Our main contribution is the vowel space plot generator prototype which produces easily understandable visual cues from analysing the biophysiological features of user speech. Our prototype is hence user-friendly for improving language learner pronunciation.

To gain evidence of our prototype being effective on assisting language learners' pronunciation training, we designed an experiment to test at the vocabulary level the feasibility and accessibility of the prototype and invited language students to provide their audio data for experimental use. Also, according to students' feedback, we proposed a series of future developments that are described in the next section. One limitation of our presented work is that there was no re-testing of pronunciation after the students received feedback from the system to check that their pronunciation improved. We plan to deploy re-tests as mentioned in our next stage experiments

7 Future Work

In the future, we aim to build on this current work to verify and quantify the pronunciation improvements gained from each user. This will help us to understand the effectiveness of this current design of the prototype and enable us to select appropriate extensions to enhance L2 learning experiences.

We are currently considering to build a correction subsystem for pronunciation practice. In addition to the existing vowel space plots, we theorise that it would be helpful to construct a system that could directly compare our users' speech to a set of externally stored standard references. This should enable the users to correct their pronunciation with higher precision and efficiency. Such a design could also potentially provide personalised pronunciation assistance via analysing user-specific pronunciation patterns.

Future iterations also intend to test a much more varied selection of MSA words that capture both short and long vowels in word initial, medial and final positions, as well as the two MSA dipthongs /aw/ (e.g. ضوء /ḍaw/ 'light') and /aj/ (e.g. بيت /bajt/ 'house') and MSA consonant.

Another potential future direction is to animate the tongue motion. Iribe et al. (2012) showed that such animations could achieve better results than their static counterparts. We expect the animated version of the vowel space plot to display tongue motions while people speak to help users to better conceptualise pronunciation in real-time.

8 Clarification: MSA Vocabulary Selection

The justification for the selection of the above ten words was based on a variety of factors. First, the selected vocabulary items were basic MSA words chosen in consultation with an MSA teacher to ensure students had been explicitly taught or otherwise been exposed to them during the course of their language learning.

Second, the selected words were restricted to one-to-three syllabic words only. This restriction ensured that sentence-level factors affecting the articulation of vowels were excluded (e.g. /t/-insertion rule in *Iḍāfah* structures; ساعة /sāʿa/ "clock" vs. ساعة يوسف /sāʿa**t** jusif/ "Joseph's clock"), thus allowing for a straightforward assessment of how the prototype detected speech boundaries and extracted the relevant features from vowel segments.

Finally, the ten words selected captured the three, cardinal MSA vowels: /a/ i/ and /u/. Although these vowels exist in the English phonemic inventory and do not theoretically pose a challenge for English-speaking L2 learners of MSA, when they are considered alongside surrounding MSA consonants then their articulation becomes more difficult, such as in the well-known case of emphatic spreading caused by the presence of pharyngeal or pharyngealised consonants ('emphatics') (e.g. Shosted et al., 2018).

Acknowledgement

The authors express their gratitude to participants and other contributors of this study. Furthermore, we would like to thank our three anonymous ALTA reviewers for their careful comments, which helped us to improve this present work.

We would also like to thank Ms Leila Kouatly, a MSA lecturer who works at the Australian National University (ANU) for helping us on the selection of the MSA glossary. She also provided us a series of opportunities to join her classes and tutorials. We acquired many valuable observations on her pedagogical methods and skills. Her activity in promoting our study ensured that students actively participated in our student experience survey and preliminary evaluation experiments.

Moreover, we thank Dr Emmaline Louise Lear and Mr Frederick Chow. Dr Lear helped us to acquire ethic approval for our study and provided us inspirations from an educator's perspective. Mr Chow helped us on communication with ANU Centre for Arab and Islamic Studies which is crucial for our study and commented on engineering details of our project. They also provided insightful suggestions for an early presentation for this study as examiners. We would like to express our sincere appreciation for their help and remarkable work.

Finally, we acknowledge the funding and support by Australian Government Research Training Program Scholarships and ANU for the first three authors' higher degree research studies.

References

Pierre Badin, Yuliya Tarabalka, Frédéric Elisei, and Gérard Bailly. 2010. Can you 'read' tongue movements? evaluation of the contribution of tongue display to speech understanding. *Speech Communication*, 52:493–503.

Steven Boll. 1979. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(2):113–120.

Judy Breitkreutz, Tracey M Derwing, and Marian J Rossiter. 2001. Pronunciation teaching practices in canada. *TESL Canada journal*, pages 51–61.

John Burgess and Sheila Spencer. 2000. Phonology and pronunciation in integrated language teaching and teacher education. *System*, 28(2):191–215.

Xinyuan Chao. 2019. *Supporting students' ability to speak a foreign language intelligibly using educational technologies:The case of learning Arabic in the Australian National University*. College of Engineering and Computer Science, The Australian National University, Canberra, ACT, Australia.

Tracey M. Derwing and Murray J. Munro. 2005. Second language accent and pronunciation teaching: A research-based approach. *TESOL Quarterly*, 39(3):379–397.

Dorina Dibra, Nuno Otero, and Oskar Pettersson. 2014. Real-time interactive visualization aiding pronunciation of english as a second language. In *2014 IEEE 14th International Conference on Advanced Learning Technologies*, pages 436–440.

Jonás Fouz-González. 2015. Trends and directions in computer-assisted pronunciation training. *Investigating English Pronunciation Trends and Directions*, pages 314–342.

Dzikri Fudholi and Hanna Suominen. 2018. The importance of recommender and feedback features in a pronunciation learning aid. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 83–87, Melbourne, Australia. Association for Computational Linguistics.

Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. 2004. Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

Yurie Iribe, Takurou Mori, Kouichi Katsurada, Goh Kawai, and Tsuneo Nitta. 2012. Real-time visualization of english pronunciation on an ipa chart based on articulatory feature extraction. *Interspeech 2012*, 2:1270–1273.

Priscilla Kan John, Emmaline Lear, Patrick L'Espoir Decosta, Shirley Gregor, Stephen Dann, and Ruonan Sun. 2020. Designing a visual tool for teaching and learning front-end innovation. *Technology Innovation Management Review*, 10.

William F. Katz and Sonya Mehta. 2015. Visual feedback of tongue movement for novel speech sound learning. *Frontiers in Human Neuroscience*, 9:612.

Shao-Hsuan Lee, Jen-Fang Yu, Yu-Hsiang Hsieh, and Guo-She Lee. 2015. Relationships between formant frequencies of sustained vowels and tongue contours measured by ultrasonography. *American journal of speech-language pathology / American Speech-Language-Hearing Association*, 24:739–749.

John Levis. 2007. Computer technology in teaching and researching pronunciation. *Annual Review of Applied Linguistics*, 27:184.

Philip Lieberman and Sheila E. Blumstein. 1988. *Speech Physiology, Speech Perception, and Acoustic Phonetics*. Cambridge Studies in Speech Science and Communication. Cambridge University Press.

Zhen-Hua Ling, Korin Richmond, and Junichi Yamagishi. 2010. An analysis of hmm-based prediction of articulatory movements. *Speech Communication*, 52(10):834–846.

Shrikanth Narayanan, Krishna Nayak, Sungbok Lee, Abhinav Sethy, and Dani Byrd. 2004. An approach to real-time magnetic resonance imaging for speech production. *The Journal of the Acoustical Society of America*, 115(4):1771–1776.

Ambra Neri, Catia Cucchiarini, Helmer Strik, and Lou Boves. 2002. The pedagogy-technology interface in computer assisted pronunciation training. *Computer Assisted Language Learning*, 15(5):441–467.

Matthias Odisio, Gérard Bailly, and Frédéric Elisei. 2004. Tracking talking faces with shape and appearance models. *Speech Communication*, 44:63–82.

Annu Paganus, Vesa-Petteri Mikkonen, Tomi Mäntylä, Sami Nuuttila, Jouni Isoaho, Olli Aaltonen, and Tapio Salakoski. 2006. The vowel game: Continuous real-time visualization for pronunciation learning with vowel charts. In *Advances in Natural Language Processing*, pages 696–703, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. 2007. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77.

Hasso Plattner, Christoph Meinel, and Ulrich Weinberg. 2009. *Design-thinking*. Springer.

Lawrence Rabiner and Ronald Schafer. 2010. *Theory and Applications of Digital Speech Processing*, 1st edition. Prentice Hall Press, Upper Saddle River, NJ, USA.

Antoine Serrurier and Pierre Badin. 2008. A three-dimensional articulatory model of the velum and nasopharyngeal wall based on mri and ct data. *The Journal of the Acoustical Society of America*, 123:2335–55.

Jia-lin Shen, Jeih-weih Hung, and Lin-shan Lee. 1998. Robust entropy-based endpoint detection for speech recognition in noisy environments. In *Fifth international conference on spoken language processing*.

Ryan K Shosted, Maojing Fu, and Zainab Hermes. 2018. *Arabic pharyngeal and emphatic consonants*, chapter chapter3. Routledge.

R. C. Snell and F. Milinazzo. 1993. Formant location from lpc analysis data. *IEEE Transactions on Speech and Audio Processing*, 1(2):129–134.

Maureen Stone. 2005. A guide to analysing tongue motion from ultrasound images. *Clinical Linguistics & Phonetics*, 19(6-7):455–501. PMID: 16206478.

Tomoki Toda, Alan W Black, and Keiichi Tokuda. 2008. Statistical mapping between articulatory movements and acoustic spectrum using a gaussian mixture model. *Speech Communication*, 50(3):215–227.

Nancy Tye-Murray, Karen Iler Kirk, and Lorianne Schum. 1993. Making typically obscured articulatory activity available to speech readers by means of videofluoroscopy. In *NCVS Status and Progress Report*, volume 4, pages 41–63.

Marla Tritch Yoshida. 2018. Choosing technology tools to meet pronunciation teaching and learning goals. *The CATESOL Journal*, 30(1):195–212.

Lingyun Yu, Jun Yu, and Qiang Ling. 2018. Synthesizing 3d acoustic-articulatory mapping trajectories: Predicting articulatory movements by long-term recurrent convolutional neural network. In *2018 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4.

Lingyun Yu, Jun Yu, and Qiang Ling. 2019. Bltrcnn-based 3-d articulatory movement prediction: Learning articulatory synchronicity from both text and audio inputs. *IEEE Transactions on Multimedia*, 21(7):1621–1632.

ABSA-Bench: Towards the Unified Evaluation of Aspect-based Sentiment Analysis Research

Abhishek Das
School of Computer Science
The University of Adelaide
abhishek.das@student.adelaide.edu.au

Wei Emma Zhang
School of Computer Science
The University of Adelaide
wei.e.zhang@adelaide.edu.au

Abstract

Aspect-Based Sentiment Analysis (ABSA) has gained much attention in recent years. ABSA is the task of identifying fine-grained opinion polarity towards a specific aspect associated with a given target. However, there is a lack of benchmarking platform to provide a unified environment under consistent evaluation criteria for ABSA, resulting in the difficulties for fair comparisons. In this work, we address this issue and define a benchmark, ABSA-Bench[1], by unifying the evaluation protocols and the pre-processed public datasets in a Web-based platform. ABSA-Bench provides two means of evaluations for participants to submit their predictions or models for online evaluation. Performances are ranked in the leader board and a discussion forum is supported to serve as a collaborative platform for academics and researchers to discuss queries.

1 Introduction

Aspect-based sentiment analysis (ABSA) has gained a lot of attention in recent years from both industries and academic communities as it provides a more practical solution to real life problems. The goal of ABSA is to identify the aspects and infer the sentiment expressed for each aspect. For example, given a sentence *I hated their service, but their food was great*, the sentiment polarities for the aspect *service* and *food* are negative and positive respectively. Conventional techniques for ABSA are mostly traditional machine learning models based on lexicons and syntactic features (Jiang et al., 2011; Kiritchenko et al., 2014; Vo and Zhang, 2015). Therefore, the performance of such models depend on hand-crafted features. Recent progresses have been made with the advancement of Deep Neural Networks (DNN) with some of the models being considered as state-of-the-art (Xu et al.). Among them, attention mechanism has played an important role outperforming previous approaches by paying more attention to the context words that are semantically-closer with the aspect terms (Luong et al., 2015; Wang et al., 2016; Chen et al., 2017; Liu et al., 2018; Ma et al., 2017). The most recent approaches adopted pre-trained Bidirectional Encoder Representations from Transformers (BERT) architecture (Devlin et al., 2019; Xu et al.) generating significant performance gaps to other approaches due to BERT's capability of capturing bi-directional contextual information and providing rich token-wise representation. Introducing BERT architecture into ABSA task naturally distinguishing the approaches to Non-BERT based models and BERT-based models. Figure 1 depicts the general processes of both of the two groups of supervised ABSA methods.

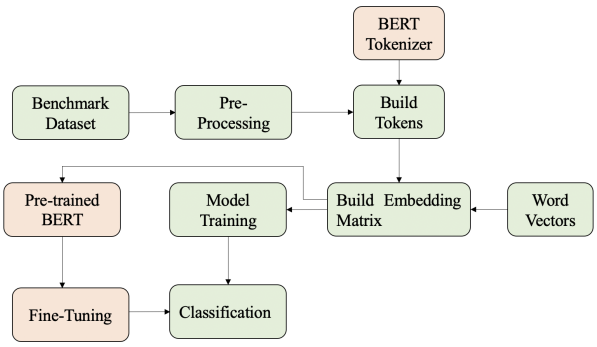


Figure 1: The General Process of ABSA

Although this research area has gained much attention in recent years, it lacks of unbiased comparisons overall. As deep learning based models perform differently on various hardware on different deep learning tools, existing works typically chose to either re-run or re-implement the selected comparative models under their own experimental environment. We also observe few works directly referring the results presented in the corresponding

[1] https://absa-bench.com/

papers for comparison. This makes it difficult to have a general overview of the performances of the state-of-the-art models and has motivated us to build a benchmarking platform for ABSA research.

Existing benchmarking research works are mostly conducted on evaluating single tasks and none of them support aspect-based sentiment analysis (Rajpurkar et al., 2016, 2018; Choi et al., 2018; Wang et al., 2019; Aguilar et al., 2020; Zhu et al., 2018). In this project, we fill this gap by proposing a unified evaluation process and building a united platform for comparing different ABSA models. We name our work as ABSA-Bench. ABSA-Bench particularly focuses on supervised approaches and is suitable for both DNN-based and conventional models. It provides two means of evaluations namely, Results Evaluation and Model Evaluation. Results evaluation is done by comparing the ground-truth with the model-generated predictions submitted by the researchers. Model evaluation supports the model submission and online evaluation which keeps the integrity of the predictions in a better way. To aid the model evaluation, a Web based tool is developed to provide an objective evaluation environment. The background computation power of ABSA-Bench is supported by the Google Cloud Platform (GCP)[2]. After evaluation, the performance results are then ranked in the ABSA-Bench leader board. ABSA-Bench further supports a discussion forum for queries, comments and discussions regarding the model implementations, performances, ranking and new ideas.

To the best of our knowledge, this is the first platform created with diverse functionalities to support the understanding of the state-of-the-art ABSA works. The contributions of the work includes: i) providing a unified ABSA evaluation platform which enables researchers to evaluate their models on the same benchmark dataset with a consistent metric under the same computation environment; ii) supporting a leader board for easy comparison, and a discussion forum for sharing ideas; iii) presenting the comparisons of several recent research works based on their performances on the ABSA-Bench platform through a re-run or re-implementation.

2 Related Works

The related benchmarking platforms for natural language processing models can be categorized into two groups: single task benchmarks and multiple tasks benchmarks. SQuAD (Rajpurkar et al., 2016, 2018)[3] is a representative benchmark for a single task. It provides a platform for evaluating question answering models on the SQuAD dataset. Researchers could either submit the prediction results or their models which will be run on CodaLab Worksheets[4]. A leader board ranks the performances of all the evaluated models. QuAC (Choi et al., 2018)[5] imitates SQuAD, but for context-aware question answering models for which the questions and answers are provided in the dialogue form. GLUE (Wang et al., 2019)[6] provides a collection of tools for evaluating the natural language understanding models across a diverse set of existing tasks. It allows researchers to submit their prediction files for comparison. Error analysis is also enabled. LinCE (Aguilar et al., 2020)[7] is a centralized benchmark for linguistic code-switching evaluation that combines ten corpora covering four different code-switched language pairs and four sub-tasks. Similar to GLUE, LinCE enables result submission, but does not support online model execution. TextGen (Zhu et al., 2018) is a benchmarking platform to support research on open-domain text generation models. It implements a majority of text generation models and aims to standardize the research in this field. However, TextGen does not allow online submission and evaluation.

ABSA-Bench is the most akin to SQuAD but unlike SQuAD, it focuses on ABSA task. ABSA-Bench provides two means of evaluations that is similar to SQuAD and QuAC. The online evaluation in ABSA-Bench is supported by JupyterHub which has key features like customization, flexibility and scalability. This distinguishes it from other similar platforms. JuptyerHub also serves a variety of environments. It can be easily containerised with any container, therefore can be scaled up for a greater number of users. A number of authentication protocols such as OAuth and GitHub are also supported, making it flexible for users. ABSA-Bench also supports an online discussion forum for researchers to exchange their ideas.

There are relatively less research efforts on providing a comprehensive benchmarking platform for multiple NLP tasks. DecaNLP (McCann et al.,

[2]https://cloud.google.com/

[3]https://rajpurkar.github.io/SQuAD-explorer/

[4]https://worksheets.codalab.org/

[5]http://quac.ai/

[6]https://gluebenchmark.com/

[7]https://ritual.uh.edu/lince/home

2018)[8] is the only one found in this category. It spans ten NLP tasks and recasts these tasks as question answering over a context using automatic transformations. Therefore, DecaNLP evaluates the models under the rubrics of assessing question answering models. DecaNLP considers the general sentiment analysis, but does not include ABSA.

3 Taxonomy and the Models

Aspect based sentiment analysis is a fundamental task in sentiment analysis research field (Pontiki et al., 2014) which comprises of three sub-tasks: aspect extraction, sentiment extraction and aspect based sentiment classification. In recent years, deep neural network has gained a lot of attention in solving the problem of ABSA. More recently, BERT (Devlin et al., 2019), has shown its effectiveness to alleviate the effort of feature engineering and has shown state-of-the art results in the given task. However these performance improvements have been achieved at a high computational cost. As a result these models are costly to train and evaluate. To have a better understanding of the large number of DNN based ABSA models, a categorization is utmost essential. Therefore, a taxonomy has been designed in this study which categorises different deep learning supervised technique, diving all approaches into broadly two categories: BERT based and Non-BERT based models. Note that we focus on supervised approaches in this work.

3.1 Models

Although the platform is designed for researchers to evaluate their models per their own need, we examined some representative models as examples.

3.1.1 Non-BERT based Models

CNN. We adopt a **C**onvolution **N**eural **N**etwork model (Xue and Li, 2018) based on convolution operations and gating mechanisms to represent the CNN-based ABSA models.

LSTM. A vanilla **L**ong **S**hort **T**erm **M**emory network represents the vanilla RNN-based models.

TD-LSTM. Target-**D**ependent **LSTM** (Tang et al., 2016a) is a modified LSTM. It consists of two LSTMs, which models the preceding and subsequent contexts surrounding the target words (aspect terms) respectively so that the contexts in both directions can be used as the feature representations for classifying sentiment in later stage.

[8] https://decanlp.com/

TC-LSTM. Target-**C**onnection **LSTM** (Tang et al., 2016a) extends TD-LSTM by adding target connection component in order to capture the interactions between target word and its contexts. This component is basically a concatenation of word embedding and target vector at each position.

ATAE-LSTM. The **AT**tention-based LSTM with **A**spect **E**mbedding (Wang et al., 2016) model appends the aspect embedding into each word input vector to capture aspect information. To capture the inter-aspect dependencies, the aspect-focused sentence representations are fed into another LSTM to model the temporal dependency.

CABASC. Content **A**ttention **B**ased **A**spect based **S**entiment **C**lassification model (Liu et al., 2018) improves the attention mechanism with the help of two attention enhancing mechanisms, i.e., sentence-level content attention and context attention. This ensures that the model is capable of taking the word order information, the aspect information and the correlation between the word and the aspect to calculate the attention weight and embed them into a series of customized memories.

IAN. Interactive **A**ttention **N**etwork considers attention mechanisms on both the aspect and the context (Ma et al., 2017). It uses two attention-based LSTM which interactively capture the key aspect terms and the important words of its context. The final representation of the sentence is produced by concatenating the representations of the aspect and its context, and is then passed to a soft-max layer for sentiment classification.

MemNet. A **Mem**ory **Net**work-based model (Tang et al., 2016b) adopts an attention mechanism with multi-hop layers which are stacked to select abstractive evidences from an external memory.

RAM. The **R**ecurrent **A**ttention mechanism based on **M**emory network (Chen et al., 2017) targets the cases that aspect terms are distant from the corresponding sentiment information. RAM introduces multiple attentions to distill the related information from its position-weighted memory and a recurrent network for sentiment classification.

3.1.2 BERT based Models

BERT-SPC. In this model, a pre-trained BERT model was fine-tuned with just one additional layer (Devlin et al., 2019). For down-stream task like ABSA, the input representation is able to represent both a single sentence and a pair of sentences.

AEN-BERT. The **A**ttentional **En**coder **N**etwork (Song et al., 2019) is built upon a BERT embedding layer along with an attentional encoder layer and a target-specific attention layer.

LCF-BERT. In this model (Zeng et al., 2019), a Local Context Focus (LCF) mechanism is proposed for aspect-based sentiment classification based on multi-head self-attention . It utilizes the Context Features Dynamic Mask and Context Features Dynamic Weighted layers to assign more attention weights to the local context words. A BERT-shared layer is adopted to capture the internal long-term dependencies of local context and global context.

BERT-PT. The BERT **P**ost-**T**raining (Xu et al.) work enhances the performance of fine-tuning of BERT for Review Reading Comprehension (RRC) by adding a post-training step. This approach was then generalised to perform the task of aspect extraction and aspect sentiment classification in aspect-based sentiment analysis.

4 The ABSA-Bench

This section introduces the ABSA-Bench platform, including the two ways of ABSA benchmarking evaluations provided and our insights into the design and implementation of ABSA-Bench.

4.1 Evaluating the Results

To evaluate the model's performance, we provide a way for researchers to submit their prediction results on the formatted test set to ABSA-Bench. The submission file needs to follow the structure required by ABSA-Bench, which is simply the sentence ID and aspect terms along with the predicted sentiment polarity. We also make available an evaluation script that we will use for the official evaluations. The evaluation script will measure the model performance based on Macro $F1$ score, which is the weighted average of Precision and Recall. It is usually a more useful accuracy measure when there is an uneven class distribution which was the case in our benchmarking dataset.

4.2 Evaluating the Models

The other means of evaluation supported by ABSA-Bench is model evaluation. We provide a unified online computation environment for researchers to train and test their models. We used widely-adopted JuypterHub[9] to which researchers could submit their model as a Juypter Notebook file.

[9]https://jupyter.org/hub

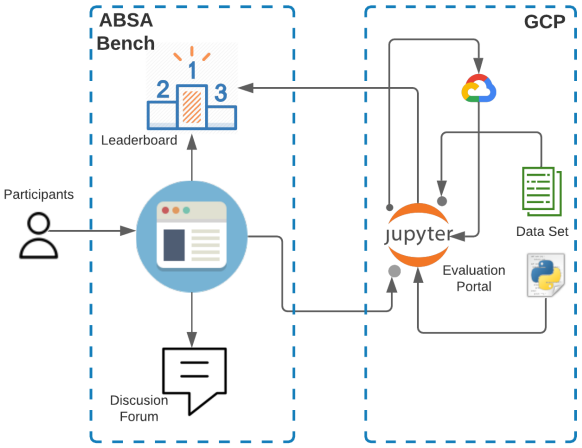


Figure 2: The Framework of ABSA-Bench

Once the trained model is submitted, it will get official scores on the test set. The platform also provides a documentation to help researchers understand how to use the platform. Please refer to Section 4.3.2 for more details.

4.3 The Web-based Platform

In order to enable the above-mentioned evaluations, we design and implement a Web-based benchmark platform that enables researchers to evaluate their ABSA models in a unified environment for fair comparison. The performances measured in Macro F-1 score is ranked in the leader board in the platform with a discussion board provided to exchange ideas among researchers. Specifically, the platform consists of three primary elements: *Leader board*, *Evaluation Portal*, and *Discussion forum*. Figure 2 shows these three elements in this platform.

4.3.1 Leader Board

We maintain a leader board in ABSA-Bench based on the evaluations of some of the state-of-the art ABSA models so far. The performances of the models that are submitted by the authors will be added to the leader board and assigned a proper ranking position. For a fair comparison, the BERT based and Non-BERT based models have been ranked separately with two tabs in the leader board.

4.3.2 Evaluation Portal

The computation power is supported by Google Cloud Platform which will serve the JupyterHub that is integrated with our platform. A pre-configured environment dedicated to ABSA will be created for participants. This environment will support complex computations and provide a task bundle which contains necessary dependencies for

the task and the evaluations. Users need to create an account and be authenticated to participate in the challenge. They can train and evaluate their model in their own work spaces leveraging the resources provided and managed by system administrators who can test the submitted prediction files and assess the submitted models under a unified standard.

4.3.3 Discussion forum

A discussion forum is provided for participants once they create their account.

This will serve as a collaborative environment where researchers can post queries and collaborate. It will be especially helpful for new academics making an initial start in this field. This will save immense time in resolving concerns through a collaborative effort.

5 Performance Comparison

Discussion on the dataset including the motivation for choice, the implementation settings for the experiments and an objective comparison of the results have been presented in this section.

5.1 Data

We adopted SemEval14 Task 4 (Pontiki et al., 2014) as the benchmarking dataset. This is because it is the only widely accepted benchmark dataset for ABSA and has successfully fostered ABSA research since its release. Although later SemEval competitions also contain ABSA tasks, those datasets are derived from the SemEval14 version with small updates that deviate the evaluation purpose from ABSA. Therefore, we retain the original version intending to be more focused.

In SemEval14 ABSA task 4, there are two domain-specific subsets for laptops and restaurants reviews respectively, consisting of over 6,000 sentences with aspect-level human-authored labels for evaluation. Each single or multi-word aspect term is assigned one of the following polarities based on the sentiment that is expressed in the sentence towards it: positive, negative, neutral, and conflict. *Restaurants* includes annotations for coarse aspect categories, aspect terms, aspect term- specific polarities, and aspect category-specific polarities. *Laptop* includes annotations for aspect terms and their polarities. We removed the data with conflict sentiment polarity and the ones without aspect terms, obtaining 1,978 training samples and 600 test for *Restaurants* and 1,462 training samples and 411 test samples for *Laptop* respectively.

Models	*Restaurants*	*Laptop*
CNN	60.25	57.75
LSTM	65.51	55.35
TD-LSTM	68.98	61.87
TC-LSTM	66.72	61.11
ATAE-LSTM	63.72	58.47
CABASC	**68.02**	**62.94**
IAN	65.12	60.90
RAM	66.76	59.73
MemNet	61.09	58.01
AEN-BERT	73.76	76.31
BERT-PT	76.96	75.08
BERT-SPC	73.03	72.63
LCF-BERT	**81.74**	**79.59**

Table 1: Performances Comparison (F-1 in %) on the Unified Environment

5.2 Implementation Adjustment

We evaluated some of the state-of-the-art ABSA models as introduced in Section 3.1. To provide a unified computation environment, we made necessary adjustments and expect researchers to follow these adjustments and submit their models to ABSA-Bench for fair comparisons.

For Non-BERT-based models, GloVe[10] is adopted as the pre-trained word embedding. We have uniformly adjusted the dimension of the hidden state vectors as 300 and position embedding as 100. We initialised the weight matrices with the uniform distribution $U(-0.1, 0.1)$, and the biases were initialised to zero. We experimented with a couple of optimizers and finally selected Adam for all the models to maintain uniformity. We kept the learning rate as $2e-5$ and used $1e-5$ as the value of the L_2 regularisation parameter.

For BERT-based models, we used a pre-trained BERT[11] model to generate word vectors of sequences. All the models were implemented using Pytorch framework. Optimal parameters were selected during the training stage and the best performed models were selected for evaluation. We kept the default settings for other parameters as set in the original papers of each work.

5.3 Results

We report the evaluation results in this section, including prediction performance, run-time statistics and model sizes comparisons.

Table 1 reports the Macro $F1$ score in % of the examined models. We have compared BERT based models and Non-BERT based models separately as BERT based models have larger model sizes.

[10]https://nlp.stanford.edu/projects/glove/
[11]https://github.com/google-research/bert

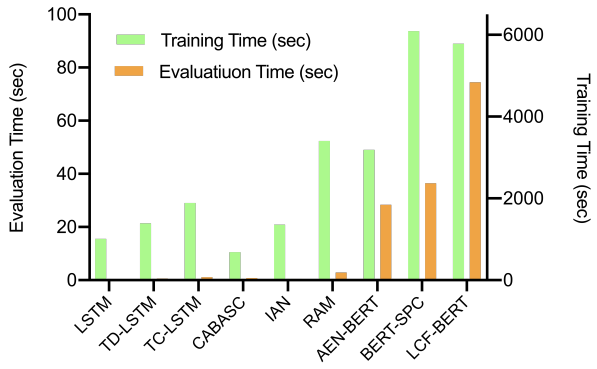


Figure 3: Model Run-Time Comparison

Models	Params 10^6	Memory (MB)
CNN	1.21	10.01
LSTM	7.23	35.61
TD-LSTM	1.44	12.41
TC-LSTM	2.16	14.11
ATAE-LSTM	2.53	16.61
CABASC	1.53	12.61
IAN	2.16	16.18
RAM	6.13	31.18
MemNet	0.36	7.8 2
AEN-BERT	112.93	451.84
BERT-PT	110	450.23
BERT-SPC	109.48	450.58
LCF-BERT	113.61	452.62

Table 2: Mode Size Comparison

BERT-based models achieved a much higher $F1$ score in comparison to Non-BERT based models as did for all the other NLP tasks. LCF-BERT model provided the best performance among BERT based models in our experiments. Among all the Non-BERT base models, CABASC has obtained the highest $F1$ score on both datasets. TC-LSTM outperforms basic LSTM model The results confirm that the context attention mechanism is more effective than the position attention mechanism. IAN outperforms ATAE-LSTM as it not only models the context representation, but also models the aspect representation by using attentions mechanism.

Figure 3 illustrates the comparisons of the model run-time i.e training and evaluation time. Table 2 present the comparisons of the model sizes in terms of the number of parameters and the size of the memory used during model training. From Figure 3 and Table 2, we observe the huge differences in the model sizes and execution times between BERT-based and Non-BERT based models. It is worth noting that for our experiments and also in the original papers, pre-trained BERT models have been used and therefore the model run time signifies time taken for fine-tuning and down-streaming the BERT model for particular task.

5.4 Evaluation Discussion

Difference in the performances. Compared to the values provided by the original papers, the performances of the examined models under our benchmarking environment ABSA-Bench show different macro $F1$ scores for all the models. It is easy to understand that the differences are as results of the different data pre-processing, implementation settings and evaluation environment. However, it is difficult to compare the models by just referring the papers. For example, the Macro $F1$ value for RAM is 70.51% for *Laptop* in (Li et al.) while the Macro $F1$ value for RAM is 71.35% for the same dataset in (Zeng et al., 2019). Given a new model with 71.00% Macro $F1$ on *Laptop*, we could not know whether it is better than RAM or not. This inconsistency motivates us to build an evaluation process on under a unified settings. Our platform aims to overcome these inconsistencies.

Trade-off between the performances and the computational costs. While BERT based models overall performed much better than Non-BERT based models, it is computationally more expensive. Even though pre-trained BERT models were used in the experiments, there was a significant increase in the computational cost which was mainly due to the huge difference in the parameter size. These models also limits research to industrial or big-scale research labs while researchers without the access to large-scale computation will be constrained with their experiments.

6 Conclusion and Future work

In this work, we design and implement an ABSA benchmarking evaluation process by providing two means of online evaluations and a Web-based platform. Leader board and discussion forums are enabled to rank the state-of-the-art ABSA research and share research ideas respectively. We examined some recent models and compared their actual differences under the unified platform ABSABench. This platform will help to understand the implementation of different deep learning models performing the task of ABSA. This understanding can then be utilised to improve the existing models. We intend to update our benchmarking platform with new tasks and datasets which will encourage quantitatively-informed research and learning.

Acknowledgments

This project is sponsored by Google Academic Research Grants.

References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation. In *Proc. of the LREC 2020*, pages 1803–1813, Marseille, France.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In *Proc. of the EMNLP 2017*, pages 452–461, Copenhagen, Denmark.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proc. of the EMNLP 2018*, pages 2174–2184, Brussels, Belgium.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the NAACL-HLT 2019*, pages 4171–4186, Minneapolis, MN, USA.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter Sentiment Classification. In *Proc. of the ACL HLT*, pages 151–160, Portland, Oregon, USA.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In *Proc. of the SemEval 2014*, pages 437–442, Dublin, Ireland.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. Transformation networks for target-oriented sentiment classification. In *Proc. of the ACL 2018*.

Qiao Liu, Haibin Zhang, Yifu Zeng, Ziqi Huang, and Zufeng Wu. 2018. Content Attention Model for Aspect Based Sentiment Analysis. In *Proc. of the WWW 2018*, pages 1023–1032, Lyon, France.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proc. of the EMNLP 2015*, pages 1412–1421, Lisbon, Portugal.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive Attention Networks for Aspect-Level Sentiment Classification. In *Proc. of the IJCAI 2017*, pages 4068–4074, Melbourne, Australia.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The Natural Language Decathlon: Multitask Learning as Question Answering. *CoRR*, abs/1806.08730.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proc. of the SemEval 2014*, pages 27–35, Dublin, Ireland.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proc. of the ACL 2018*, pages 784–789, Melbourne, Australia.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *Proc. of the EMNLP 2016*, pages 2383–2392, Austin, USA.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Targeted sentiment classification with attentional encoder network. In *Proc of the ICANN 2019*, pages 93–103.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective LSTMs for Target-Dependent Sentiment Classification. In *Proc. of the COLING 2016*, pages 3298–3307, Osaka, Japan.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect Level Sentiment Classification with Deep Memory Network. In *Proc. of the EMNLP 2016*, pages 214–224, Austin, Texas, USA.

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Proc. of the IJCAI 2015*, pages 1347–1352, Buenos Aires, Argentina.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE:A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proc. of the ICLR 2019*, New Orleans, LA, USA.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In *Proc. of the EMNLP 2016*, pages 606–615, Austin, Texas.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In *Proc. of the NAACL-HLT 2019*, Minneapolis, MN, USA.

Wei Xue and Tao Li. 2018. Aspect Based Sentiment Analysis with Gated Convolutional Networks. In *Proc. of the ACL 2018*, pages 2514–2523, Melbourne, Australia.

Biqing Zeng, Heng Yang, Ruyang Xu, Wu Zhou, and Xuli Han. 2019. LCF: A Local Context Focus Mechanism for Aspect-Based Sentiment Classification. *Applied Sciences*, 9:3389.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A Benchmarking Platform for Text Generation Models. In *Proc. of the SIGIR 2018*, pages 1097–1100, Ann Arbor, MI, USA.

A machine-learning based model to identify PhD-level skills in job ads

Lian Chen[1], Hanna Suominen[1-4] and **Inger Mewburn[1]**
[1]The Australian National University (ANU)/ Canberra, ACT, Australia
[2]Data61, Commonwealth Scientific and Industrial Research Organization (CSIRO/ Canberra, ACT, Australia)
[3]University of Canberra/ Canberra, ACT, Australia
[4]University of Turku/ Turku, Finland

Abstract

Around 60% of doctoral graduates worldwide ended up working in industry rather than academia. There have been calls to more closely align the PhD curriculum with the needs of industry, but an evidence base is lacking to inform these changes. We need to find better ways to understand what industry employers really want from doctoral graduates. One good source of data is job advertisements where employers provide a 'wish list' of skills and expertise. In this paper, a machine learning-natural language processing (ML-NLP) based approach was used to explore and extract skill requirements from research intensive job advertisements, suitable for PhD graduates. The model developed for detecting skill requirements in job ads was driven by SVM. Our preliminary results showed that ML-NLP approach had the potential to replicate manual efforts in understanding job requirements of PhD graduates. Our model offers a new perspective to look at PhD-level job skill requirements.

1 Introduction

Abundant evidence shows that industry employers are often dissatisfied with key aspects of PhD graduates' workplace performance (e.g., Cumming, 2010; G08, 2013; Australian Department of Education, 2014; Hancock, 2019), particularly in relation to professional skills like communications (McCarthy & Wient, 2019). PhD graduates themselves also indicate the education they received during their candidature did not address job market needs outside of academia (Golde & Dore, 2001). Reports indicate that employers were often dissatisfied with doctoral employees' demonstration of soft skills at work (e.g., Cumming, 2010; Cyranoski et al., 2011). Such frustration from stakeholders leads us to question the fitness for purpose of doctoral degrees.

PhD was originally designed to help people into academic careers, but its fitness for purpose has been questioned for over 80 years (Dale, 1930). As the number of PhD graduates increase tremendously (Auriol et al., 2013; Gould, 2015) and the academic job market remains relatively static in scale (Larson et al., 2014), many PhD graduates will be unable to secure academic positions. Despite the fact that most PhD graduates will be working outside academia, universities are still training their candidates based on research competencies desired in academia. One such example of a popular research training framework is the Vitae Researcher Development Framework (RDF) based on research by Bray and Boon (2011). To enhance PhD employability, Mewburn et al. (2018) argue it is not enough to understand only academic workforce requirements: non-academic professions may have different needs.

Some initiatives have been undertaken to understand the so called 'transferrable skills' needed from PhD job seekers. Consequently, many add-on courses based on the long list of skill terms have been put in place at universities (Barnacle & Mewburn, 2010). However, scholars such as Neumann and Tan (2011) and Platow (2012) have expressed concerns about the generic quality of these initiatives. Take professional skills such as teamwork and empathy: these may mean very different things in different workplaces and industry domains. Concerns are reasonable about the ambiguity of previous initiatives suspicious of over-generalisation of skills and neglect of the context in which skills are deployed. However, little has been done so far to empirically test the difference that context makes. We can ask employers to tell us, but relying on retrospective self-reports has an inherent problem of informant inaccuracy (Bernard et al., 1984; Ellison et al., 2020). Other methods such as ethnography are too difficult to scale up considering the large number of industry fields. Therefore, a Machine Learning / Natural Language (ML/NLP) approach is worthy of exploration.

To address the gap described above, we developed a machine-learning-based model to identify employers' expectations of qualified job seekers in job adverts. We first manually labelled 400 job ads across two industries based on Move-Step analysis, an analytic approach widely adopted by applied linguistics researchers to detect contextual difference in written language discourse (Bhatia, 2014). Moves

are coarse-grained categories, and steps are fine-grained categories associated with a particular move. Finer granularity is essential to avoiding abstractness and ambiguity in language use. Hence, a data retrieval interface with fine- and coarse-grained information can provide more accurate results than an interface without differentiation in information granularity (Zhang et al., 2020). After the manual annotation was done, the labelled data were fed into the machine using the SVM algorithm. After the parameters were tuned, the majority of the identified skill categories in the model reached good Area Under the Curve (AUC) performance. Although the model remains to be optimised for certain skill categories, the results showed that natural language processing of job advertisements has the potential to replicate human efforts to provide rich insights into how PhD education can be improved.

This paper seeks to contribute to the literature in three key ways:

1. Workshop design: We outline how we developed a coders' workshop for this project. Our workshop experience can be valuable for future attempts to use natural language processing to inform higher education policy making.
2. Curriculum design tools: The coarse and fine granularity in our annotation generates more accurate definitions of skill items, which in turn enables better curriculum design in PhD education.
3. Evidence: Our model can serve as a useful approach to testify the hypothesis that contextual difference exists across industry domains. In addition, it gives evidence of automation being a feasible approach to contribute to human efforts in understanding PhD-level skill requirements in job advertisements by accelerating abilities to analyse text systematically at scale.

2 Background and related work

Job ads contain rich information regarding employers' expectation of qualified job seekers (Walker & Hinojosa, 2014), yet research on the automated identification of skills from job ads is still in its infancy. Most previous studies on skills in job ads were manual content analysis (see for example, Pitt & Mewburn, 2016). The majority of studies on job ads only examined a single industry domain. Studies with manual efforts examined professions such as librarian positions (Clyde. 2002) and accounting (Tan & Laswad, 2018). Nevertheless, studies relying on machine learning techniques seem to exclusively examine computing related job positions (e.g., Aken et al., 2009; Ericsson & Wingkvist, 2014; Khaouja et al., 2018; Rahhal et al., 2019). In recent years, scholars have started analysing and extracting skills in job ads with machine learning techniques. The effort to automatically retrieve skill requirements from job ads should enable a content analysis approach to job ads to be expanded to other industry domains.

Although there are several ML-based studies on job ads, most authors only took technical skills into consideration. For example, Ericsson and Wingkvist (2014), Khaouja et al. (2018), Rahhal et al. (2019) identified technical skills such as programming languages but ignored non-technical skills such as teamwork, communication skills, user engagement, workplace aesthetics, ethics, etc. Obviously, non-technical skills in job ads remain underexplored by data mining researchers. In order to have a holistic understanding of employers' expectations, we need to examine both technical and non-technical skills listed in job ads.

3 Methods

3.1 Analytic framework

At the manual annotation stage, Move-step analysis was adopted. This analytic framework, first proposed by Swales (1990), has been widely adopted by applied linguistics researchers in exploring conventions in written discourse of communities of practice[1] (Bhatia, 2014; Moreno & Swales, 2018). In Move-step analysis, researchers take the stance that a particulate genre reflects social habitus[2] of a community of practice through events and goals they record in their texts (Bhatia, 2014). When the context of a text genre changes, these components vary in quantity or quality (Connor, 2000; Maswana et al., 2015). A most straightforward example is research articles. The structural components of research articles in different scientific disciplines are not entirely consistent. In a similar vein, job ads across industry domains could differ in textual conventions.

Moves in a genre are the overarching communicative purposes which can be achieved through alternative steps (Swales, 1990). This hierarchical differentiation of information granularity is very useful for users of results of information retrieval tasks (Tange et al., 1998; Fonseca et al., 2002; Zhang et al., 2020). As move step analytic framework is an inductive approach, it allows enough autonomy for annotators to minimise influence from predefined categories which are likely to be vaguely defined (Neumann & Tan, 2011). Such autonomy is particularly important when most categories of skills in existent reports, according to Platow (2012), do not derive from

[1] The term 'communities of practice', coined by Wenger (1999), denotes groups of people who share the same goals, interests, and knowledge.

[2] 'Social habitus' refers to habits, dispositions and skills someone has as a result of immersing in a social environment (Bourdieu, 1986).

solid theoretical justification[3].

Below is an example of the move 'Continuous education' and its associated steps identified from the job ad data in this study:

Move: Continuous education

- **Step 1**: Passion & Self-motivation
  Example: *A passion for developing web-based applications.*
- **Step 2**: Participation in training
  Example: *In return we'll offer training, scope for progression and a support buddy to keep you company while you get your feet under the desk.*
- **Step 3**: Sharing of knowledge
  Example: *…recommend corrective actions advice.*
- **Step 4**: Seeking advice
  Example: *Continuously seek(s) feedback and responds proactively.*
- **Step 5**: Self-reflection
  Example: *Demonstrate the ability to identify and support practice improvements and support the implementation of best practice.*

3.2 Data

Research intensive job ads were chosen to explore employability skills required of doctoral graduates. The raw data were purchased from the job market solution company Burning Glass Technologies Inc. Ethics approval was obtained for using the data for our research purpose. Although the purchased data cannot be shared due legal and ethical concerns, future scholarly attempts to verify the results from this study can rely on job ad data from the public domain. One example of using publicly available job ads for analysis is Pitt and Mewburn (2016). Alternatively, scholars interested in accessing the same data could contact Burning Glass Technologies Inc directly. Our intention of purchasing data was for scaling up the analysis across industries in the future.

The raw data were further filtered into research skill intensive job ads via the algorithm developed by the PostAc® team (Mewburn et al., 2018; Xu et al., 2019) at the Australian National University. The PostAc® filter reached good performance of above 80% accuracy.

Overall, 400 high research skill intensive job ads posted from 2015 to 2016 were randomly chosen for manual annotation. The total word counts of the dataset were 147,089. Of the 400 job ads, 200 were targeting healthcare job seekers, and another 200 were targeting computing job seekers. The reason we chose healthcare and computing professions is because these two domains have the greatest potential to grow in the next five years (Australian Government, 2019). The healthcare dataset consisted of 74,179 of word frequencies. The computing dataset comprised 72,910 of word frequencies.

Several steps were taken to pre-process the data in preparation for the machine learning experiments. We first segmented the data into sentences as units for annotation. This process helped us see if a skill was mentioned several times in one job ad, which possibly indicates the importance of the skill for the employer. If we had treated a whole job ad as an analytic unit, a skill requirement could have been only counted once even if it is mentioned several times. We also removed stop words such as articles and conjunctions from the machine-readable dataset via the stop-word list in the NLTK v3.5 corpus. For the training, testing and validation purposes, the labelled dataset was separated into 70%, 15% and 15% of the overall dataset accordingly.

3.3 The chosen algorithm

The chosen algorithm for running the experiment in the study is Support Vector Machine (SVM) (Cortes & Vapnik, 1995) with the linear kernel. An analytic unit could contain multiple skill requirements. Hence our task is a multilabel text classification task. SVM is an algorithm widely adopted to deal with multilabel text classification tasks (Qin & Wang, 2009; Yang et al., 2009; Wang & Chiang, 2011). Another reason for choosing linear SVM is because it is a computationally cost-effective algorithm which at the same time guarantees good prediction outcome for text classification tasks (Vijayan et al., 2017). It is worth mentioning that we also piloted using Naive Bayes and Logistic Regression classifiers potentially suitable for multi-label text classification tasks with the default parameters. SVM is the one that obviously outperformed these piloted baselines on our dataset. In the experiment, SVM parameters were tuned for optimization using the GridSearchCV [4] tool. The parameters tuned are listed as follows:

- Max-iteration,
- Loss,
- Tolerance,
- Fit intercept, and
- Intercept scaling.

The classifiers' performance was evaluated using the Area Under the Curve (AUC). The AUC measure can avoid the problematically 'too good' results derived from the situation where the accuracy score is high, but the evaluation is biased by class imbalance (Suominen et a., 2009; Narkhede, 2018).

[3] Platow (2012) points out that authors of reports which provide a list of skills did not specify the context in which the skills were deemed important.

[4] The GridSearchCV tool was taken from scikit-learn v0.23.2

3.5 Coders' workshop

One of the authors and a research assistant were involved in our Coders' workshop. Both annotators hold masters' degrees and were experienced in annotation tasks.

In the coders' workshop, we agreed that several aspects listed as follows are essential to ensuring the quality of the annotation.

1. It is important to have more than one annotator to do the annotation independently. This is to ensure the inter-coder reliability. Our intercoder reliability at step level reached 0.76 measured by Cohen's Kappa[5]. As achieving a good intercoder reliability at the step level is a direct indication that move-level units were labelled reliably, we only calculated Kappa at the step level.

2. Compared to one-off pilot annotation, it is more reasonable to hold several rounds of discussion on improvements among annotators in between annotation efforts. In other words, an iterative process of continuous improvement would be better.

3. There should be a mechanism to resolve interpersonal conflict when disagreement occurs between annotators. In our case, we marked down dubious items in our notes during the discussion before we continue in the next round of annotation with an eye for evidence and justification for our opinions.

4 Results and discussion

Overall, 12 skills at move level and their associated steps were identified from our manual analysis (see table 1). The fine-grained steps in our model serve as a tool to unpack the meaning of coarse-grained 'umbrella terms', which in the past were considered by scholars as having little information regarding their contextual interpretation (Barnacle & Mewburn, 2010). The results and discussion comprise two parts. The first part is the report on machine learning performance. The second part is the report on Chi-square test for skill categories with good ML performance.

4.1 Machine learning performance

The machine learning experiment results in table 1 showed that the AUC scores of many step categories on training, test and validation sets are close, which indicates the model is very likely to produce similar results on unseen data.

It can also be seen from table 1 that 28 out of 61 step-level categories reached AUC scores above 0.8 on all training, validation, and test sets. These categories account for 46% of the overall step-level categories.

Table 1. AUC results of the experiment

Moves	Steps	Train	Val	Test
Empathy with	Clients	0.98	0.89	0.90
	Children	0.98	0.89	0.92
	Clients' family members	0.97	0.87	0.83
	Aged group	0.97	0.89	0.83
	Ethnic minorities	0.98	0.87	0.82
	General public	0.88	0.61	0.66
	Novices	0.91	0.79	0.74
	Disabled group	0.98	0.97	0.88
	LGBTIQ+ community[6]	0.98	1.00	1.00
	Women	0.97	0.85	0.50
People skills	Network with peers	0.93	0.77	0.80
	Interpersonal skills	0.96	0.84	0.82
	Multidisciplinary collaboration	0.96	0.87	0.83
	Network with decision makers	0.89	0.63	0.68
	Network with partners	0.93	0.78	0.74
	Network with the public	0.85	0.50	0.50
	Network with public sectors	0.95	0.67	0.65
	Network with private sectors	0.92	0.63	0.50
	Network with research institutions	0.80	0.80	0.79
	Network with project sponsors	0.75	0.75	0.73
	Network with research participants	0.95	0.59	0.69
Continuous education	Participate in training	0.94	0.86	0.84
	Passion & Self-motivation	0.93	0.83	0.80
	Share knowledge	0.92	0.84	0.80
	Seek advice	0.85	0.71	0.68
	Self-reflection	0.98	0.69	0.70
Cognitive abilities	Analytic skills	0.92	0.77	0.76
	Needs extraction	0.95	0.78	0.78
	Understand problems	0.98	0.94	0.88
	Innovation	0.98	0.88	0.90
Professional standards	Ethical conduct	0.67	0.60	0.64
	Policy & Regulation	0.96	0.88	0.82
	Background-check	0.98	0.93	0.96
	Confidentiality	0.95	0.79	0.85
Personal attributes	Personal impact	0.85	0.80	0.79
	Leadership skills	0.95	0.84	0.83
	Result orientation	0.94	0.72	0.75
	High-pressure management	0.91	0.73	0.67

[5] Cohen's Kappa equation: $K = \frac{P(a)-P(e)}{1-P(e)}$ Where P(a) denotes observed percentage of agreement, and P(e) denotes the probability that agreement is met by chance. Cohen's Kappa works for assessing categorical variables (Hallgren, 2013), and hence is suitable for this study.

[6] The step 'Empathy with LGBTIQ+ community' has relatively simpler features. When an analytic unit contains the key words of 'LGBT', 'sexuality' or 'gay', the model would very possibly predict it as this 'Empathy with LGBTIQ+ community'. We therefore agreed that the this is the reason why the AUC results for both Val and Test sets are 1.

	Attention to safety	0.98	0.92	0.88
	Commercial orientation	0.88	0.65	0.73
	Time efficiency	0.94	0.76	0.84
	Independence	0.94	0.77	0.82
	Attention to details	0.93	0.73	0.75
Aesthetics	Maintain workplace	0.96	0.84	0.82
	Manage configuration	0.98	0.83	0.83
	Manage resource	0.95	0.72	0.75
Courage	Work in harsh environment	0.91	0.60	0.65
	Manage conflicts	0.98	0.66	0.77
	Manage risks	0.95	0.83	0.86
	Manage change	0.97	0.80	0.89
	On-call availability	0.92	0.50	0.67
	Driving & Travelling	0.97	0.85	0.80
Proof of qualification	Register in institutions	0.93	0.83	0.81
	Writing skills	0.94	0.79	0.69
	Attain tertiary degree	0.93	0.83	0.88
	General IT skills	0.95	0.70	0.69
	Industry experience	0.96	0.84	0.84
	Oral presentation	0.90	0.80	0.75
	Residency	0.98	0.83	0.96
None category	*The category not containing any skill requirement	0.96	0.89	0.90
Healthcare technical skills		0.88	0.68	0.70
Computing technical skills		0.92	0.82	0.78

Although not all the skill categories reached the rule of thumb gold standard of 0.8 in AUC performance, the results from the experiment so far indicate a likelihood that the manual efforts can be replicated by the machine when further optimization is conducted.

Specifically, step categories such as 'Empathy with the general public' which reached an AUC score above 0.9 on training set but below 0.8 on validation and test sets are likely to have the overfitting problem. The linear SVM is an algorithm that already is less prone to overfitting (Baumes et al., 2006). Hence, future attempts to avoid overfitting for optimization include increasing the number of training samples in order for the machine to capture the key features of the category.

For categories (e.g., the step 'Network with project sponsors') which reached an AUC score below 0.8 on training, validation and test sets, it is very likely that the quality of the manual annotation is not good enough for the machine to learn well. Such underfitting problem can be avoided when future attempt to improve the manual coding procedure is conducted.

Due to the necessity of further optimizing the model, it is still not feasible for our model to be used immediately in another study. Nevertheless, our model can be used to identify those 28 step categories which reached AUC score above 0.8 on all training, test, and validation sets in PhD-level healthcare and computing job ads.

4.2 Chi-square test for step categories with good AUC scores

Previously, there was little empirical evidence to testify the assumption about contextual difference in skill requirements. We therefore did chi-square test for the 27 step-level skill categories (excluding 'None category') with good AUC performance based on our manually labelled data set. The results suggest that there is a significant difference between the two industry domains in 22 of these 27 step-level categories. The Chi-square test results for these categories are listed in table 2 below.

Table 2. Chi-square test results for the 27 steps with good ML performance

Steps	***Comp***	***Health***	***P value***	X^2
Empathy with clients	625	602	> .5	0.4
Empathy with children	8	226	< .0001	203.1
Empathy with clients' family members	5	175	< .0001	160.6
Empathy with aged group	5	143	< .0001	128.7
Empathy with ethnic minorities	101	159	< .001	12.9
Empathy with disabled group	2	124	< .0001	118.1
Empathy with LGBTIQ+ community	7	30	< .001	14.3
Interpersonal skills	367	383	> .5	0.3
Multidisciplinar-y collaboration	106	179	< .0001	18.7
Participate in training	159	221	< .01	10.1
Passion & Self-motivation	474	431	< .5	2
Share knowledge	365	391	< .5	0.9
Understand problems	311	292	< .5	0.6
Innovation	208	149	< .01	9.8
Policy & Regulation	163	452	< .0001	135.8
Background-check	115	213	< .0001	29.3
Leadership skills	199	143	< .01	9.2
Attention to safety	67	170	< .0001	44.8
Maintain workplace	16	145	< .0001	103.4
Manage configuration	82	2	< .0001	76.2
Manage risks	164	99	< .0001	16.1
Manage change	188	123	< .001	13.6
Driving & Travelling	61	70	> .5	0.6

Register in institutions	4	279	< .0001	267.2
Attain tertiary degree	112	181	< .0001	16.3
Industry experience	157	102	< .001	11.7
Residency	70	80	< .5	0.7

There is no significant difference in the occurrences of 'Empathy with clients' and 'Interpersonal skills' between computing and healthcare industries. Both industries required qualified job seekers to maintain positive relationships with people at work. Nevertheless, job ads in healthcare industry required job seekers to empathize with a wider range of communities than in computing industry. Healthcare job seekers need to have stronger capacity of emphasizing with children, clients' family members, the aged group, ethnic minorities, people with disability, and the LGBTIQ+ community. Such difference in 'Empathy' might indicate greater subtlety and complexity of healthcare professionals' workplace interpersonal relationships.

'Multidisciplinary collaboration' in healthcare job ads was mentioned more frequently in computing job ads. This difference might be because the healthcare professionals often need to deal with complex health problems beyond one's specialization (Vissers et al., 2013). In comparison, computing professionals receive everyday tasks whose technical scope are already pinpointed. Possibly it is because of a weaker multidisciplinary orientation that computing professionals were required more often to understand problems by one's own (Gardner, 2010), as indicated by more frequent mentioning of the step 'Understand problems' in computing (shown in table 2).

There was only slight difference between the two industries in the steps of 'Participate in training', 'Passion & Self-motivation', 'Share knowledge', 'Innovation', and 'Leadership skills'. Whereas computing job seekers were required more often to have 'Passion & Self-motivation', 'Leadership skills' and 'Innovation', healthcare job seekers were asked more often to 'Share knowledge' and 'Participate in training'. Since computing industry frequently experiences innovation and drives change in technology (as indicated by the step 'Manage change' and 'Innovation' in table 2), professionals in this field naturally need to have stronger passion and self-motivation to keep pace. Also, problems in the field are sometimes ill-structured (Brown, 2008), and hence the ability to lead and manage change in clients' needs and technology plays a significant role in computing industry.

The more frequent requirement in healthcare industry of 'Policy & Regulation', 'Background-check', 'Attention to safety', 'Maintaining workplace' and 'Managing risks' (table 2) could be associated with the patient-related uncertainties in healthcare context. As healthcare professionals' practices are directly relevant to patients' physical and mental wellbeing, familiarization with policies, regulations, safety guidelines, risk prevention is crucial in order to cope with potential hazards and disputes.

The step 'Manage configuration' seemed to be a requirement specific to the computing industry, as shown in table 2. Configuration is domain-specific resource that computing practitioners need to manage in their daily work. According to Stevenson (2010), configuration in computing refers to the set-up of software and hardware components of a product. Hence, configuration is a special resource in computing that is different from other resource such as those in the healthcare industry.

Through Chi-square test, we pinpointed difference in the number of skills required in computing and healthcare industries. These difference challenges the view that doctoral graduates' identity is monolithic, especially under the current circumstance where more than 60% PhDs ended up working outside of academia (Larson et al., 2014). As suggested by Gardner, (2010), doctoral students' identity formation, skill development and socialization should be linked with specific industrial and professional contexts when devising support in their programs. In this study, we illustrate how it would be problematic to offer add-on courses and de-contextualized interpretation of skills to PhD students. Analysis of job ads in different industries, and machine-learning-enabled automation of the analysis are potential methods to enable better decision making in PhD education.

5 Conclusion

In this study, we developed a model to automatically identify some PhD-level skill requirements in job ads. The move-step analysis we adopted in our manual annotation procedure allows for the model to pinpoint both coarse-grained skill items and their associated contextual interpretation. The moves and steps we identified can be used in curriculum design to enhance PhD candidates' awareness of skill requirements in different industry domains. Our human coder's workshop experience can be useful for scholars who also intend to conduct ML driven analysis of textual data for enabling better decision making in higher education. In addition, the ability of our model to quantify skills provides evidence that contextual difference exists in the number of skills required of qualified PhD job seekers. Our finding challenges the problematic view that we can set aside contextual factors in PhD training.

Our model has limitations. In this study, we pinpointed areas for further optimizing the model. Before the model can be used to automatically identify all the skill categories, the overfitting and underfitting problems need to be solved. Additionally, we only experimented with limited number of data in healthcare

and computing domains. Therefore, extra effort to manually label more data in these two and other industries is necessary before the model can be used to identify skills across different industry contexts.

Acknowledgement

We are grateful for the support from Burning Glass Technologies Inc, PostAc®, and ANU CV Discovery Translation Fund2.0. Our thanks also go to the anonymous paper reviewers and Dr. Will Grant for their insightful comments on this paper. We thank Dr. Lindsay Hogan and Chenchen Xu for offering us advise on the technical and legal requirements involved in this study. We appreciate the anonymous annotator's contribution in our coders' workshop. Finally, the first author would like to thank Australian Government Research Training Program International Scholarship for supporting her PhD studies.

References

Aken, Andrew., Litecky, Chuck., Ahmad, Altaf., & Nelson, Jim. (2009). Mining for computing jobs. *IEEE software*, *27*(1), 78-85.

Auriol, Laudeline., Misu, Max., & Freeman, R. Ann. (2013, March). Careers of doctorate holders: Analysis of labour market and mobility indicators. Retrieved from https://www.oecd-ilibrary.org/docserver/5k43nxgs289w-en.pdf

Australian Department of Education (2014, June). *Initiatives to enhance the professional development of research students*. Retrieved from https://docs.education.gov.au/system/files/doc/other/initiatives_to_enhance_the_prof_development_of_research_students_0.pdf

The Australian Government. (2019). *Future Outlook.*

Barnacle, Robyn., & Mewburn, Inger. (2010). Learning networks and the journey of 'becoming doctor'. *Studies in Higher education*, *35*(4), 433-444.

Baumes, L. A., Serra, J. M., Serna, P., & Corma, A. (2006). Support vector machines for predictive modeling in heterogeneous catalysis: a comprehensive introduction and overfitting investigation based on two real applications. *Journal of combinatorial chemistry, 8*(4), 583-596.

Bernard, H. Russell., Killworth, Peter., Kronenfeld, David., & Sailer, Lee. (1984). The problem of informant accuracy: The validity of retrospective data. *Annual review of anthropology*, *13*(1), 495-517.

Bhatia, Vijay. (2014). *Analysing genre: Language use in professional settings*. London: Routledge.

Bourdieu, Pierre. (1986) 'The Forms of Capital', in J.G. Richardson (ed.) *Handbook of Theory and Research for the Sociology of Education* (pp. 241-59). New York: Greenwordpress.

Bray, Robert., & Boon, Stuart. (2011). Towards a framework for research career development. *International Journal for Researcher Development*.

Brown, T. (2008). Design thinking. *Harvard business review*, *86*(6), 84.

Clyde, Laurel. A. (2002). An instructional role for librarians: an overview and content analysis of job advertisements. *Australian academic & research libraries*, *33*(3), 150-167.

Connor, Ulla. (2000). Variation in rhetorical moves in grant proposals of US humanists and scientists. Retrieved from https://scholarworks.iupui.edu/handle/1805/2663

Cortes, Corinna., & Vapnik, Vladimir. (1995). Support-vector networks. *Machine learning, 20*(3), 273-297.

Cumming, Jim. (2010). Contextualised performance: Reframing the skills debate in research education. *Studies in Higher Education*, *35*(4), 405-419.

Cyranoski, David., Gilbert, Natasha., Ledford, Heidi., Nayar, Anjali., & Yahia, Mohammed. (2011). The PhD factory: The world is producing more PhDs than ever before. Is it time to stop?. *Nature*, *472*(7343), 276-279.

Dale, Edgar. (1930). The Training of Ph. D.'s. *The Journal of Higher Education*, *1*(4), 198-202.

Ellison, William. D., Trahan, Alec. C., Pinzon, Joseph. C., Gillespie, Megan. E., Simmons, Logan. M., & King, Kendel. Y. (2020). For whom, and for what, is experience sampling more accurate than retrospective report? *Personality and Individual Differences*, *163*, 10071.

Ericsson, Morgan., & Wingkvist, Anna. (2014, June). *Mining job ads to find what skills are sought after from an employers' perspective on IT graduates*. [Paper presentation]. 2014 conference on Innovation & technology in computer science education, New York, US. https://dl.acm.org/doi/10.1145/2591708.2602670

Fonseca, Frederico., Egenhofer, Max., Davis, Clodoveu., & Câmara, Gilberto. (2002). Semantic granularity in ontology-driven geographic information systems. *Annals of mathematics and artificial intelligence*, *36*(1-2), 121-151.

G08 (2013, March). *The Changing PhD: Discussion paper*. Retrieved from https://go8.edu.au/files/docs/the-changing-phd_final.pdf

Gardner, Susan. K. (2010). Contrasting the socialization experiences of doctoral students in high-and low-completing departments: A qualitative analysis of disciplinary contexts at one institution. *The Journal of Higher Education*, *81*(1), 61-81.

Golde, Chris. M., & Dore, Timothy. M. (2001). At cross purposes: What the experiences of today's doctoral

students reveal about doctoral education. Retrieved from https://files.eric.ed.gov/fulltext/ED450628.pdf

Gould, Julie. (2015). How to build a better PhD. *Nature News*, *528*(7580), 22.

Hallgren, Kevin. A. (2012). Computing inter-rater reliability for observational data: an overview and tutorial. *Tutorials in quantitative methods for psychology*, *8*(1), 23.

Hancock, Sally. (2019). A future in the knowledge economy? Analysing the career strategies of doctoral scientists through the principles of game theory. *Higher Education*, *78*(1), 33-49.

Khaouja, Imane., Rahhal, Ibrahim., Elouali, Mehdi., Mezzour, Ghita., Kassou, Ismail., & Carley, Kathleen. M. (2018, April). Analyzing the needs of the offshore sector in Morocco by mining job ads. [Paper presentation]. *2018 IEEE Global Engineering Education Conference (EDUCON)*, Tenerife,Spain. https://ieeexplore.ieee.org/document/8363390

Larson, Richard. C., Ghaffarzadegan, Navid., & Xue, Yi. (2014). Too many PhD graduates or too few academic job openings: the basic reproductive number R0 in academia. *Systems research and behavioral science*, *31*(6), 745-750.

Maswana, Sayako., Kanamaru, Toshiyuki., & Tajino, Akira. (2015). Move analysis of research articles across five engineering fields: What they share and what they do not. *Ampersand*, *2*, 1-11.

McCarthy, Paul., & Wient, Maaike. (2019). *Who are th e top PhD employers*: *Advancing Australia's know ledge economy*. Retrieved from https://blogs.deak in.edu.au/bl-research-student-news/wp-content/up loads/sites/227/2019/09/Advancing_Australias_K nowledge_Economy-0021.pdf

Mewburn, Inger., Grant, Will. J., Suominen, Hanna., & Kizimchuk, Stephanie. (2018). A machine learning analysis of the non-academic employment opportunities for Ph. D. graduates in Australia. *Higher Education Policy*, 1-15.

Moreno, Ana. I., & Swales, John. M. (2018). Strengthening move analysis methodology towards bridging the function-form gap. *English for Specific Purposes*, *50*, 40-63.

Narkhede, Sarang. (2018). Understanding AUC-ROC Curve. *Towards Data Science*, *26*.

Neumann, Ruth., & Tan, Kim. K. (2011). From PhD to initial employment: the doctorate in a knowledge economy. *Studies in Higher Education*, *36*(5), 601-614.

Pitt, R., & Mewburn, I. (2016). Academic superheroes? A critical analysis of academic job descriptions. *Journal of Higher Education Policy and Management*, *38*(1), 88-101.

Pitt, Rachael., & Mewburn, Inger. (2016). Academic superheroes? A critical analysis of academic job descriptions. *Journal of Higher Education Policy and Management*, *38*(1), 88-101.

Platow, Michael. J. (2012). PhD experience and subsequent outcomes: A look at self-perceptions of acquired graduate attributes and supervisor support. *Studies in Higher Education*, *37*(1), 103-118.

Qin, Yu. P., & Wang, Xiu. K. (2009, August). *Study on multi-label text classification based on SVM*. [Pap er presentation]. 2009 Sixth International Confere nce on Fuzzy Systems and Knowledge Discovery, Tianjin, China. https://ieeexplore.ieee.org/abstra ct/document/5358597

Rahhal, Ibrahim., Makdoun, Ibtissam., Mezzour, Ghita., Khaouja, Imane., Carley, Kathleen., & Kassou, Ismail. (2019, April). *Analyzing Cybersecurity Job Market Needs in Morocco by Mining Job Ads*. [Paper presentation]. 2019 IEEE Global Engineering Education Conference *(EDUCON)*, Dubai, United ArabEmirates. https://ieeexplore.ieee.org/document/8725033

Stevenson, Angus. (Ed.) (2010). Oxford dictionary of English (3rd ed.). Oxford: Oxford University Press.

Suominen, H., Pyysalo, S., Hiissa, M., Ginter, F., Liu, S., Marghescu, D., Pahikkala, T., Back, B., Karsten, H., Salakoski, T. (2008). Performance evaluation measures for text mining. In M. Song & Y-FB. Wu (Eds.), *Handbook of Research on Text and Web Mining Technologies* (pp. 724–747). Pennsylvania, USA: IGI Global.

Swales, John. M. (1990). *Genre analysis: English in academic and research settings*. Cambridge: Cambridge University Press.

Tan, Lin. M., & Laswad, Fawzi. (2018). Professional skills r-equired of accountants: what do job adve rtisements tell us?. *Accounting Education*, *27*(4), 403-432.

Tange, Huibert. J., Schouten, Harry. C., Kester, Arnold. D., & Hasman, Arie. (1998). The granularity of medical narratives and its effect on the speed and completeness of information retrieval. *Journal of the American Medical Informatics Association*, *5*(6), 571-582.

Vijayan, Vikas. K., Bindu, K. R., & Parameswaran, Lat -ha. (2017, September). *A comprehensive study of text clas-sification algorithms.* [Paper presentatio n]. 2017 International Conference on Advances in Computing, Communications and Informatics (IC ACCI), Udupi, India. https://ieeexplore.ieee.org/ document/8125990

Vissers, Kris. C., van den Brand, Maria. W., Jacobs, Jose., Groot, Marieke., Veldhoven, Carel., Verhagen, Constans., Hasselaar, Jeroen., & Engels, Yvonne. (2013). Palliative medicine update: a multidisciplinary approach. *Pain Practice*, *13*(7), 576-588.

Walker, H. Jack., & Hinojosa, Amanda. S. (2014). Recruitment: The Role of Job Advertisements. In K. Yu & D. M. Cable (Eds.), *The Oxford*

handbook of recruitment (pp. 269-283). New York: Oxford University Press.

Wang, Tai. Y., & Chiang, Huei. M. (2011). Solving multi-label text categorization problem using support vector machine approach with membership function. *Neurocomputing*, *74*(17), 3682-3689.

Wenger, Etienne. (1999). *Communities of practice: Learning, meaning, and identity*. Cambridge: Cambridge university press.

Xu, Chenchen., Mewburn, Inger., Grant, Will. J., & S uominen, Hanna. (2019, July). *PostAc: A Visual Interactive Search, Exploration, and Analysis Pl atform for PhD Intensive Job Postings*. [Paper pr esentation]. The 57th Annual Meeting of the Ass ociation for Computational Linguistics: System Demonstrations, Florence, Italy. https://www.acl web.org/anthology/P19-3008.pdf

Yang, Bishan., Sun, Jian. T., Wang, Tengjiao., & Chen, Zheng. (2009, June). *Effective multi-label active l earning for text classification*. [Paper presentation]. The 15th ACM SIGKDD international conferen ce on Knowledge discovery and data mining, New York, US. https://dl.acm.org/doi/10.1145/15570 19.1557119

Zhang, Min., Li, Feng., Wang, Yang., Zhang, Zequan., Zhou, Yanhai., & Li, Xiaoyu. (2020). Coarse and Fine Granularity Graph Reasoning for Interpretab le Multi-Hop Question Answering. *IEEE Access* , *8*, 56755-56765.

Learning Causal Bayesian Networks from Text

Farhad Moghimifar, Afshin Rahimi, Mahsa Baktashmotlagh and **Xue Li**
The School of ITEE
The University of Queensland, Australia
{f.moghimifar,a.rahimi,m.baktashmotlagh}@uq.edu.au
xueli@itee.uq.edu.au

Abstract

Causal relationships form the basis for reasoning and decision-making in Artificial Intelligence systems. To exploit the large volume of textual data available today, the automatic discovery of causal relationships from text has emerged as a significant challenge in recent years. Existing approaches in this realm are limited to the extraction of low-level relations among individual events. To overcome the limitations of the existing approaches, in this paper, we propose a method for automatic inference of causal relationships from human written language at conceptual level. To this end, we leverage the characteristics of hierarchy of concepts and linguistic variables created from text, and represent the extracted causal relationships in the form of a Causal Bayesian Network. Our experiments demonstrate superiority of our approach over the existing approaches in inferring complex causal reasoning from the text.

1 Introduction

Causation is a powerful psychological tool for human to choreograph his surrounding environment into a mental model, and use it for reasoning and decision-making. However, inability to identify causality is known to be one of the drawbacks of current Artificial Intelligence (AI) systems (Lake et al., 2015). Extraction of causal relations from text is necessity in many NLP tasks such as question answering and textual inference, and has attracted a considerable research in recent years (Wood-Doughty et al., 2018; Zhao et al., 2018, 2017; Ning et al., 2018; Rojas-Carulla et al., 2017). However, the state-of-the-art methods are limited to the identification of causal relations between low-level individual events (Dunietz et al., 2017; Hidey and McKeown, 2016; Mirza and Tonelli, 2016) and fail to capture such relationships at conceptual level. Furthermore, relying on linguistic features limits the identification of causal relations to those whose cause and effect are located in the same sentence or in consecutive sentences.

In this paper, we propose a method for extracting concepts and their underlying causal relations from written language. Furthermore, to leverage the extracted causal information, we represent the causal knowledge in the form of a **C**ausal **B**ayesian **N**etwork (CBN). Having this tool enables answering complex causal and counter-factual questions, such as: *How psychotherapy can affect the patient's emotion?*, or *What would happen if instead of medicine X, medicine Y was prescribed?*

The contribution of this paper is three-fold. Firstly, we focus on identifying causal relation between concepts (e.g. *physical activity* and *health*). Secondly, We propose a novel method to represent the extracted causal knowledge in the form of a Causal Bayesian Network, enabling easy incorporation of this invaluable knowledge into downstream NLP tasks. Thirdly, we release PSYCAUS dataset which can be used to evaluate causal relation extraction models in the domain of psychology [1]. In addition, our proposed method identifies causality between concepts independent of their locations in text, and is able to identify bi-directional causal relations between concepts, where two concepts have causal effect on each other. By aggregating linguistic variable, we construct a hierarchy where each variable, e.g. delusional disorder, lies under its related concept, e.g. disorder. This hierarchical and inheritance structure allows for the inference of causal relations between concepts that are not directly discussed in the text.

In order to evaluate our proposed method, we gathered a corpus of psychological articles The experimental results shows that the proposed method performs significantly better than the state-of-the-

[1] https://github.com/farhadmfar/psycaus

art methods.

2 Related Works

Identification of causality in NLP is not trivial as a result of language ambiguity. Hence, most current approaches focus on verb-verb, verb-noun, and noun-noun relations. The explicit relations are often captured with narrow syntactic and semantic constructions (Do et al., 2011; Hendrickx et al., 2009; Mirza and Tonelli, 2016; Hidey and McKeown, 2016) which limits their recall. To go beyond surface form constructions few works have proposed neural models (Martínez-Cámara et al., 2017; Dasgupta et al., 2018; Zhao et al., 2017) covering wider causal constructions. However, most works don't go beyond extracting causality between adjacent events, and so lack the ability to capture causality in non-adjacent concept level, e.g. *genetics* and *hallucination*. Therefore, in this paper we propose a model for identifying causality between concepts, independent of their location, and represent the causal knowledge in form of a Causal Bayesian Network.

3 Methodology

Given the input, in form of human written language, we aim to extract the causal relation between concepts and represent the output in form of a Causal Bayesian Network. Hence, we split this task into three sub-tasks: extracting linguistic variables and values, identifying causality between extracted variables, and creating conditional probability table for each variable. In the following sub-sections each of these sub-tasks are explained.

3.1 Linguistic Variables

A *linguistic variable* is a variable which values are words in natural language (Zadeh, 1983). For example, if we consider the word Age as a linguistic variable, rather than having numeric values, its values are linguistic, such as young and old. A linguistic word is specified as $(C, T(C))$ where C in the name which represents the set of words or in other word the variable's name, and $T(C)$ is the set of represented words or linguistic values. In this context, a variable and corresponding value have an asymmetric relation, in which the hypernym (superordinate) implies the hyponym (subordinate).

In order to create a Bayesian Network (BN) from text, we first need to extract linguistic variables and values from our corpus. To this end, we leverage a probabilistic method introduced by Wu et al. (2012) to extract all possible *IsA* relations from corpus.

To enhance the accuracy of causality identification and runtime performance of our model, using Formal Concept Analysis (Ganter and Wille, 2012), we represent the extracted hypernym-hyponym relations in form of a hierarchy. In the context of our corpus, let V be the set of linguistic variables and v be the set linguistic values, we call the triple of (V, v, I) a formal context where V and v are non-empty sets and $I \subseteq V \times v$ is the incidence of the context. The pair of (V_i, v_i) is called a formal concept of (V, v, I) if $V_i \subseteq V, v_i \subseteq v, V_i^{'} = v_i$ and $v_i^{'} = V_i$, where $V_i^{'}$ and $v_i^{'}$ are sets of all attributes common to V and v, respectively. The formal concept of a given context are naturally in form of a subconcept-superconcept relation, given for formal concepts of (V_i, v_i) and (V_j, v_j) of (V, v, I) : $(V_i, v_i) \leq (V_2, v_2) \iff V_i \subseteq V_j(\iff v_i \subseteq v_j)$. Consequently, we can identify that every attributes in the former formal concepts are also in the latter. Hence, this set of formula gives us the hierarchy of superconcept-subconcepts. Since every link in this hierarchy implies inheritance, attributes at the higher level are inherited by lower nodes. Therefore, if a concept V_i at level n of our hierarchy has a causal relation with another concept V_j, all the subconcepts of V_i at lower level m (where $m < n$), also have causal relation with V_j.

3.2 Identifying Causality

The core part of this paper is to identify the cause-effect relation between concepts, or linguistic variables. In a lower-level approach, causality is usually presented by syntactic relations, where a word or a set of words implies existence of causality. For example, 'cause' in 'Slower-acting drugs, like fluoxetine, may cause discontinuation symptoms" indicates a causal relation. These set linguistic features can be shown either in form of a verb or a discourse relation. The Penn Discourse Tree Bank (PDTB) (Prasad et al., 2008) contains four coarse-grained relations, comparison, contingency, expansion and temporal, in which contingency may indicate causality. There are 28 explicitly causal marker out of 102 in PDTB, with the barrier of causal relation. Furthermore, we leverage the sets of verbs included in `AltLexes`, such as 'force' and 'caused', which show causality in a sentence. Using both discourse and verb makers of causality,

we create a database of cause-effect (Γ) from given sentences. To this end, each of the input sentences are split into simpler version, using dependency parser, and once any of causality markers are identified in a sentence, the stopwords from cause and effect parts are eliminated and the remaining words are stemmed. Having the constructed cause-effect database (Γ), the causal relation between two concepts is defined as:

$$\mathrm{CR}(V_m, V_n) = \frac{\sum_{i=1}^{|V_m|}\sum_{j=1}^{|V_n|}\mathbb{1}[(v_m^i, v_n^j)\in\Gamma]\vec{v}_m^i\cdot\vec{V}_m}{\sum_{i=1}^{|V_m|}\vec{v}_m^i\cdot\vec{V}_m} - \frac{\sum_{j=1}^{|V_n|}\sum_{i=1}^{|V_m|}\mathbb{1}[(v_n^j, v_m^i)\in\Gamma]\vec{v}_n^j\cdot\vec{V}_n}{\sum_{j=1}^{|V_n|}\vec{v}_n^j\cdot\vec{V}_n} \quad (1)$$

where V_m and V_n are two concepts or linguistic variables in the concept space V, and v_m^i is the i-th value of V_m; the functions r and w are defined as below:

$$r(a,b) = \begin{cases} 1 & if(a,b)\in\Gamma \\ 0 & if(a,b)\notin\Gamma \end{cases} \quad (2)$$

and

$$w(a,b) = 1 - S_c(a,b) = 1 - sim(a,b) \quad (3)$$

where sim is Cosine similarity of words a and b. The purpose of function w is to measure the relevance of the value to the corresponding variable, in order to increase the influence of more relevant values. The output of CR can be categorised as follow:

$$CR(A,B) \in \begin{cases} (\mu, 1] & \text{A cause; B effect} \\ [-\mu, \mu] & \text{no causal relationship} \\ [-1, -\mu) & \text{B cause; A effect} \end{cases} \quad (4)$$

where μ is a threshold given to the model as a hyper-parameter.

3.3 Creating Conditional Probability Distribution

Conditional Probability Distribution (CPD) shows conditional probability of corresponding values to a variables with respect to values of parents of the variable $P(X_i|Parents(X_i))$. In order to extend the implementation of CPD for sets of linguistic variables we use *Normalized Pointwise Mutual Information (PMI)* score to calculate the probability distribution (Bouma, 2009).

$$i_n(x,y) = (ln\frac{p(x,y)}{p(x)p(y)})/-ln(p(x,y)) \quad (5)$$

The reason behind using *PMI* comes from Suppes' Probabilistic theory of Causality (Suppes, 1973), where he mentions that possibility of an effect to co-happen with cause is higher than happening by itself. In mathematical word it can be shown as $P(effect|cause) > P(effect)$, which can be easily written as $\frac{P(\text{cause, effect})}{P(\text{cause})P(\text{effect})} > 1$, similar to *PMI* for positive values.

To create a Causal Bayesian Network from textual data, let G be our graphical model, and $V = \{V_1, V_2, ..., V_n\}$ be the set of extracted linguistic variables (as defined in §3.1) from our corpus ζ. We define $Pa_{V_i}^G = \{Vj : Vj \xrightarrow{causal} V_i\}$, indicating set of nodes in ζ which have causal relation with V_i. By expressing P as:

$$P(V_1, V_2, ..., V_n) = \prod_{i=1}^{n} P(V_i|Pa_{V_i}^G) \quad (6)$$

we can argue that P factorises over G. The individual factor $P(V_i|Pa_{V_i}^G)$ is called conditional probability distribution (explained in 3.3). In a more formal way, we define Causal Bayesian Network over our corpus ζ as $\beta = (G, P)$ where P is set of conditional probability distributions. In addition to the aforementioned usages of a CBN, having a Causal Bayesian Networks enables the possibility of answering questions in three different layers of Association, Intervention and Counter-factual (Pearl and Mackenzie, 2018).

4 Experimental Results

In this section, we evaluate the effectiveness of our method and compare it with that of the state-of-the-art methods on a collection of Wikipedia articles related to Psychology. Each article within this collection is selected based on the terms in APA dictionary of Psychology (VandenBos, 2007). This collection contains a total number of 3,763,859 sentences. Among all possible relation between concepts, we studied 300 relationships between concepts, which were annotated by 5 graduate student from school of Psychology. Each tuple of relationship in form of (A, B), were labelled as $-1, 0$, or 1, where 1 indicates $A \xrightarrow{cause} B$, -1 shows that $B \xrightarrow{cause} A$, and 0 implies no causal relations. With the overlap tuples (25%) we measured the degree of agreement between annotators using Fleiss' Kappa measure (Fleiss and Cohen, 1973), which was around 0.67. This indicates the reliability of our test setup.

We compare our model to the feature-based and distribution-based methods proposed by Rojas-Carulla et al. (2017), with different proxy projection functions, including { *w2vii, w2vio, w2voi, counts, prec-counts, pmi, prec-pmi*}. Furthermore, we compare our model to heuristic models,

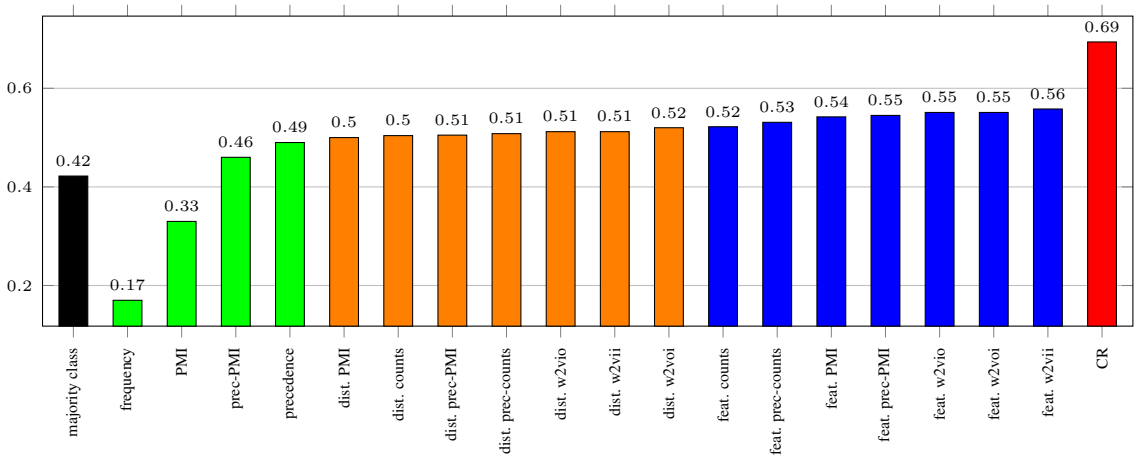


Figure 1: The test accuracy of *CR* compared with feature-based, distribution-based, heuristic methods, and the majority class.

consisting of frequency, precedence, PMI, PMI (precedence), where in each of the models two parameters are calculated, $S_{V_i \to V_j}$ and $S_{V_i \to V_j}$, indicating $V_i \xrightarrow{cause} V_j$ if $S_{V_i \to V_j} > S_{V_i \to V_j}$ and $V_i \xleftarrow{cause} V_j$ if $S_{V_i \to V_j} < S_{V_i \to V_j}$.

Figure 1 shows the accuracy of different methods for identifying causal relationships. We observe that our method (the red bar) outperforms all other approaches with an accuracy of 0.694. This indicates an improvement of 13% over the state-of-the-art feature-based methods (the blue bars), 17% over the distribution-based approaches (the orange bars), and 20% over the baseline methods (green bars). The baseline methods represented the worst performance, however, the accuracy achieved by precedence suggests that most of our corpus is written in form of active rather than passive voice, resulting in consequential connection between concepts.

To analyse the sensitivity of our method to the threshold μ in Equation 1, we trained the model on PSYCAUS's training set(D_{tr}), and analysed the development set performance in terms of macro-averaged F1 with a range of values $[0, 1)$ for μ. As shown in Figure 2, F1 score reaches the maximum of 0.66 with $\mu = 0.05$, well above a random classifier.

During the annotation process, we noticed that some concepts, e.g. *eating disorder* and *emotion*, may have bi-directional causal relations, depending on the context (eating disorder $\xleftrightarrow{\text{cause}}$ emotion). We ran our model against these examples and found out that our approach is interestingly capable of identifying these relations as well. In Equation 1, approximation of absolute values of both operands in the negation to one indicates bi-directional causality. While a bi-directional causal relation cannot be presented in a CBN, as it is a *directed* graphical model, a decision tree can contain these types of information. In addition, some concepts, e.g. *delusional disorder* and *displeasure*, that have not been connected with any type of causality connectives were also accurately identified as causal relations. This is due to the hierarchical design of variable-values in our model.

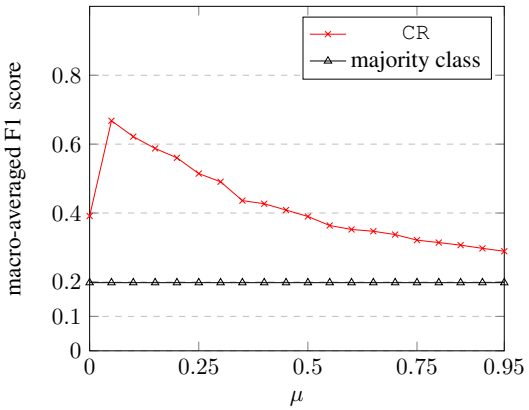


Figure 2: The macro-averaged F1 score of our proposed method on the development of PSYCAUS with different values of μ (Equation 1), compared with macro-averaged F1 score of the majority class model.

5 Conclusion

In this paper we have presented a novel approach for identifying causal relationship between concepts. This approach enables machines to extract causality even between non-adjacent concepts. Hence, a significant improvement was delivered comparing to naive baselines. Furthermore, we represented the causal knowledge extracted from human-written language in form of Causal Bayesian Network. To the best of our knowledge this representation is novel. Having a Causal Bayesian Network can empower many downstream applications, including question-answering and rea-

soning. Among all applications, causal and counterfactual reasoning, which can be build on top of the outcome of this paper, may address some current hallmarks of Artificial Intelligence systems.

References

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

Tirthankar Dasgupta, Rupsa Saha, Lipika Dey, and Abir Naskar. 2018. Automatic extraction of causal relations from text using linguistically informed deep neural networks. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 306–316.

Quang Xuan Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics.

Jesse Dunietz, Lori Levin, and Jaime Carbonell. 2017. Automatically tagging constructions of causation and their slot-fillers. *Transactions of the Association for Computational Linguistics*, 5:117–133.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.

Bernhard Ganter and Rudolf Wille. 2012. *Formal concept analysis: mathematical foundations*. Springer Science & Business Media.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics.

Christopher Hidey and Kathy McKeown. 2016. Identifying causal relations using parallel wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1424–1433.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Eugenio Martínez-Cámara, Vered Shwartz, Iryna Gurevych, and Ido Dagan. 2017. Neural disambiguation of causal lexical markers based on context. In *12th International Conference on Computational Semantics Short papers (IWCS)*.

Paramita Mirza and Sara Tonelli. 2016. Catena: Causal and temporal relation extraction from natural language texts. In *Proceedings the 26th International Conference on Computational Linguistics: Technical Papers*, pages 64–75.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 2278–2288.

Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic Books.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*. Citeseer.

Mateo Rojas-Carulla, Marco Baroni, and David Lopez-Paz. 2017. Causal discovery using proxy variables. *arXiv preprint arXiv:1702.07306*.

Patrick Suppes. 1973. A probabilistic theory of causality.

Gary R VandenBos. 2007. *APA dictionary of psychology*. American Psychological Association.

Zach Wood-Doughty, Ilya Shpitser, and Mark Dredze. 2018. Challenges of using text classifiers for causal inference. *arXiv preprint arXiv:1810.00956*.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492. ACM.

Lotfi A Zadeh. 1983. Linguistic variables, approximate reasoning and dispositions. *Medical Informatics*, 8(3):173–186.

Sendong Zhao, Meng Jiang, Ming Liu, Bing Qin, and Ting Liu. 2018. Causaltriad: Toward pseudo causal relation discovery and hypotheses generation from medical text data. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 184–193. ACM.

Sendong Zhao, Quan Wang, Sean Massung, Bing Qin, Ting Liu, Bin Wang, and ChengXiang Zhai. 2017. Constructing and embedding abstract event causality networks from text snippets. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 335–344. ACM.

Benchmarking of Transformer-Based Pre-Trained Models on Social Media Text Classification Datasets

Yuting Guo*[1], Xiangjue Dong*[1], Mohammed Ali Al-Garadi[2], Abeed Sarker[2], Cécile Paris[3], Diego Mollá-Aliod[4]
[1]Department of Computer Science, Emory University, Atlanta, GA, USA
[2]Department of Biomedical Informatics, Emory University, Atlanta, GA, USA
[3]CSIRO Data61, Sydney, Australia
[4]Department of Computing, Macquarie University, Sydney, Australia
{yuting.guo, xiangjue.dong, m.a.al-garadi, abeed.sarker}@emory.edu
cecile.paris@data61.csiro.au, diego.molla-aliod@mq.edu.au

Abstract

Free text data from social media is now widely used in natural language processing research, and one of the most common machine learning tasks performed on this data is classification. Generally speaking, performances of supervised classification algorithms on social media datasets are lower than those on texts from other sources, but recently-proposed transformer-based models have considerably improved upon legacy state-of-the-art systems. Currently, there is no study that compares the performances of different variants of transformer-based models on a wide range of social media text classification datasets. In this paper, we benchmark the performances of transformer-based pre-trained models on 25 social media text classification datasets, 6 of which are health-related. We compare three pre-trained language models, RoBERTa-base, BERTweet and ClinicalBioBERT in terms of classification accuracy. Our experiments show that RoBERTa-base and BERTweet perform comparably on most datasets, and considerably better than ClinicalBioBERT, even on health-related datasets.

1 Introduction

Transformer-based pre-trained language models have proven to be effective for many natural language processing (NLP) tasks, such as text classification and question answering, and they have enabled systems to outperform previous state-of-the-art approaches. A prime example of such language representation models is Bidirectional Encoder Representations from Transformers (BERT), which was pre-trained on the Book Corpus and English Wikipedia (Devlin et al., 2019). Since it was proposed, many efforts have attempted to improve upon it, and common strategies for doing so are to use more data and train longer (Liu et al., 2019), or to pre-train from scratch on domain-specific data (Gu et al., 2020). Multiple variants of transformer-based models have been proposed, but there is currently limited information available about how the variants directly compare on a set of similar tasks.

In this paper, we focus on text from a specific source, namely, social media, and the common task of text classification. We compare the performances of three pre-training methods. We chose text classification as our target task because it is perhaps the most common NLP-related machine learning task, and most of the publicly-available annotated datasets were prepared for it. We included 25 social media classification datasets, 6 of which are health-related. We compared three transformer-based models—RoBERTa-base (Liu et al., 2019), BERTweet (Nguyen et al., 2020a), and ClinicalBioBERT (Alsentzer et al., 2019). Our experiments show that RoBERTa-base and BERTweet perform comparably and are considerably better than ClinicalBioBERT. In addition to comparing the performances of the models on all the datasets, we analyzed the differences in performances between domain-specific (medical), source-specific (social media), and generic pre-trained models. Our empirical analyses suggest that RoBERTa-base can capture general text characteristics, while BERTweet can capture source-specific knowledge, and pre-training on large-scale source-specific data can improve the capabilities of models to capture general text features, potentially benefiting downstream source-specific tasks.

2 Related Work

The most relevant and recent related works are those by Peng et al. (2019) and Gu et al. (2020). Peng et al. (2019) proposed the Biomedical Language Understanding Evaluation (BLUE) benchmark for the biomedical domain. The evaluations include five tasks with ten datasets covering both biomedical and clinical texts. The specific tasks in-

clude named entity recognition, text classification and relation extraction. Gu et al. (2020) proposed the Biomedical Language Understanding and Reasoning Benchmark (BLURB) for PubMed-based biomedical NLP applications, with 13 biomedical NLP datasets in six tasks. To the best of our knowledge, there is no existing work that attempts to perform similar benchmarking for transformer-based approaches on social media data, and the results reported in this paper follow on the footsteps of the benchmarks referenced above.

Recent attempts at adaptation of transformer-based models are also relevant to our current work, since we wanted to include a domain-adapted and a source-adapted model in our comparisons. Many domain adaptation efforts have been reported in the literature. BioBERT—generated by pre-training BERT on biomedical corpora (*e.g.*, PubMed abstracts)—was demonstrated to outperform BERT on three representative biomedical text mining tasks (Lee et al., 2019). Alsentzer et al. (2019) attempted to further adapt pre-trained models for clinical text by training BioBERT on clinical notes, resulting in the ClinicalBioBERT model. We included ClinicalBioBERT as an example of a domain-adapted pre-trained model in our comparisons. For source-adaptation (social media text), Nguyen et al. (2020a) proposed BERTweet by pre-training BERT on a large set of English tweets. We include BERTweet in our comparisons as an example of a source-adapted model.

3 Methods

3.1 Model Architecture

We focus solely on benchmarking systems for social media text classification datasets in this paper. The overall framework of our classification model is shown in Figure 1. It consists of an encoder, a pooling layer, a linear layer, and an output layer with Softmax activation. The encoder converts each token in a document into a embedding matrix, and the pooling layer generates a document embedding e_d by averaging the word embeddings.[1] The document embedding is then fed into the linear layer and the output layer. The output is a probability value between 0 and 1, which is used to compute a logistic loss during the training phase, and the class with the highest probability is chosen in the inference phase. We use the encoders from recent pre-trained deep language models that are trained on different corpora and pre-training tasks to convert documents into embeddings, as described in Section 3.2.

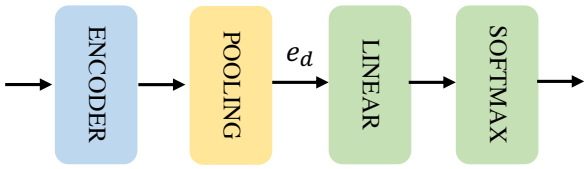


Figure 1: The overall framework of our model.

3.2 Document Encoder

RoBERTa: A BERT variant named RoBERTa was released by Liu et al. (2019) with the same model architecture of BERT but with improved performance, achieved by training the model longer with a larger batch size, on more data, removing the next sentence prediction objective during the pre-training procedure, and applying a dynamic masking technique. We chose RoBERTa-base as the generic or domain-independent encoder in this paper since it outperforms BERT-base and matches the state-of-the-art results of another BERT variant XLNet (Yang et al., 2019) on some NLP tasks.

BERTweet: Nguyen et al. (2020a) developed BERTweet, a pre-trained deep language model with the same model architecture as BERT-base, but using the RoBERTa pre-training procedure on a large scale set of English tweets. Because tweets generally use informal grammar and irregular vocabulary, which are different from traditional text data such as news articles and Wikipedia, BERTweet was an attempt at source adaptation of pre-trained models. BERTweet has been shown to obtain better results than RoBERTa-base on three Tweet NLP tasks—POS tagging, named entity recognition and text classification, illustrating its higher capability of capturing language features of English Tweets compared to RoBERTa-base (Nguyen et al., 2020a).

ClinicalBioBERT: ClinicalBioBERT (Alsentzer et al., 2019), is built by further training of BioBERT (Lee et al., 2019) on clinical notes, and it has been shown to significantly outperform BERT-base on three clinical NLP tasks. This model can generate contextual word embeddings, which are expected to capture clinical knowledge and can benefit the clinical NLP tasks such as natural language inference and entity recognition in the medical domain.

[1] We also experimented with [CLS] embeddings, but did not observe significant performance differences (Appendix A.2).

3.3 Data

We included 25 datasets in our experiments, comprising 6 datasets that were created for health-related tasks such as prescription medication abuse and adverse drug reaction detection, and 19 that were created for non-health-related tasks such as sentiment analysis and offensive language detection. The detailed data descriptions are listed in the Appendix A.1, and the statistics of all datasets are described in Table 1. For data preprocessing, we followed the procedure implemented by the open source tool *preprocess-twitter*,[2] which includes the steps of lowercasing, and normalizing numbers, hashtags, links, capital words and repeated letters.

	Dataset	TRN	TST	L	S
Health	ADR Detection	4318	1152	2	T
	BreastCancer	3513	1204	2	T
	PM Abuse	11829	3271	4	T
	SMM4H-17-task1	5340	6265	2	T
	SMM4H-17-task2	7291	5929	3	T
	WNUT-20-task2	6238	1000	2	T
Non-Health	OLID-1	11916	860	2	T
	OLID-2	11916	240	2	T
	OLID-3	11916	213	3	T
	TRAC-1-1	11999	916	3	F
	TRAC-1-2	11999	1257	3	T
	TRAC-2-1	4263	1200	3	Y
	TRAC-2-2	4263	1200	2	Y
	Sarcasm-1	3960	1800	2	R
	Sarcasm-2	4500	1800	2	T
	CrowdFlower	28707	8101	13	T
	FB-arousal-1	2085	580	9	F
	FB-arousal-2	2088	590	9	F
	FB-valence-1	2064	595	8	F
	FB-valence-2	2066	604	9	F
	SemEval-18-A	1701	1002	4	T
	SemEval-18-F	2252	986	4	T
	SemEval-18-J	1616	1105	4	T
	SemEval-18-S	1533	975	4	T
	SemEval-18-V	1182	938	8	T

Table 1: The statistics of the training (`TRN`) and test (`TST`) set. `L`: #classes; `S`: data sources; `T`: Twitter; `R`: Reddit; `F`: Facebook; `Y`: YouTube.

3.4 Experimental Setup

Following a modified setting from Liu et al. (2019), we performed a limited parameter search with learning rate $\in \{2e-5, 3e-5\}$. We fine-tuned each model for 10 epochs and selected the model that achieves the best metric on the validation set. Each experiment was run three times with different initializations, and the median results of the validation and test sets for each dataset are reported. The rest of hyper-parameters were empirically chosen and are shown in Table 2.

Hyper-parameter		Hyper-parameter	
Max sequence size	128	Warmup ratio	0
Batch size	32	Adam epsilon	1e-8

Table 2: Hyper-parameter configurations of all models.

4 Results and Discussion

Table 3 lists the accuracies of all the models on the test sets of the included datasets. In order to compare the statistical significance of differences between the accuracies, we used the McNemar's test to compare the top-2 best models for each dataset. The difference between two models is regarded as statistically significant if the p-value <0.05.

Dataset	RB	BT	CL	p-value
ADR Detection	91.4	**92.7**	90.4	0.11
BreastCancer	**93.9**	93.6	91.2	0.90
PM Abuse	81.4	**82.4**	77.4	0.09
SMM4H-17-task1	**93.6**	93.5	92.7	0.76
SMM4H-17-task2	78.4	**79.7**	75.0	**0.01**
WNUT-20-task2	**89.1**	88.3	86.5	0.48
OLID-1	85.1	**85.2**	83.5	0.90
OLID-2	89.4	**90.0**	89.0	0.73
OLID-3	69.5	**70.0**	66.4	0.73
TRAC-1-1	58.6	**59.2**	55.4	0.76
TRAC-1-2	58.8	**65.8**	58.0	**0.00**
TRAC-2-1	72.8	**73.3**	63.9	1.00
TRAC-2-2	85.8	85.5	**87.2**	0.10
sarcasm-1	67.3	**69.5**	64.6	0.06
sarcasm-2	73.2	**76.1**	68.2	**0.02**
CrowdFlower	39.9	**41.3**	38.8	**0.00**
fb-arousal-1	46.6	45.3	**46.8**	1.00
fb-arousal-2	**54.9**	54.8	54.1	0.92
fb-valence-1	60.2	**64.4**	54.5	0.06
fb-valence-2	**52.8**	52.6	45.9	1.00
SemEval-18-A	52.3	**54.6**	46.0	0.16
SemEval-18-F	**69.3**	67.4	65.3	0.09
SemEval-18-J	47.7	**51.5**	45.3	**0.01**
SemEval-18-S	**54.9**	53.9	48.4	0.42
SemEval-18-V	45.5	**46.6**	36.2	0.56

Table 3: The accuracies of the three transformer-based models on the test splits of our included datasets. `RB`: RoBERTa; `BT`: BERTweet; `CL`: ClinicalBioBERT; `p-value`: McNemar's test p-value. The best result of each dataset and the p-values <0.05 are in boldface.

BERTweet achieves the highest accuracies on 16 out of 25 datasets, including health and non-health-related datasets from Twitter, Facebook, Reddit, and YouTube. The fact that BERTweet performs well on non-tweet datasets suggests that BERTweet can learn some universal characteristics of social media languages by pre-training on tweets.[3] On 5 datasets (specifically, `SMM4H-17-task2`,

[2] https://nlp.stanford.edu/projects/glove/preprocess-twitter.rb

[3] Dai et al. (2020) reported a similar finding: a model pre-trained on business reviews (Forum BERT) outperformed one

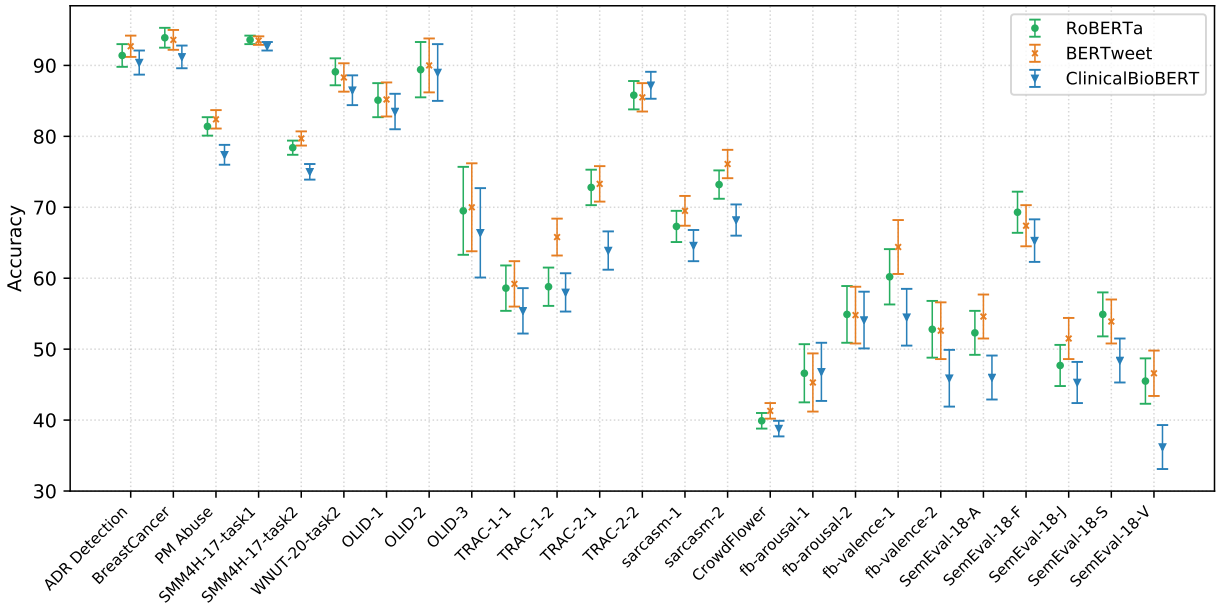


Figure 2: The 95% confidence intervals of three models on our included datasets.

`TRAC-1-2`, `sarcasm-2`, `CrowdFlower`, and `SemEval-18-J`), the best-performing system obtained significantly better results than the next best system, and in all these cases, BERTweet was the winner. There are, however, no significant differences between RoBERTa-base and BERTweet on most datasets, which shows that RoBERTa-base can capture general text features and work well on social media tasks. The differences in the pre-training dataset sizes for RoBERTa-base (160 GB) and BERTweet (80 GB) suggest that pre-training on relatively small source-specific data may effectively benefit the downstream source-specific tasks.

Figure 2 visually illustrates the distribution of the accuracy scores and their 95% confidence intervals for all three models on our included datasets. From the figure, the relative underperformance of the ClinicalBioBERT is evident. ClinicalBioBERT does not appear to capture social media-specific characteristics of the data even for health-related classification datasets, although it is trained on clinical notes. This finding suggests that for social media-specific health-related research tasks, it might be better to choose a source-specific pre-trained model (*e.g.*, BERTweet for social media) rather than a domain-specific one. It is possible that the gap between the language of clinical notes and social media text is large enough to negatively impact the social media text representation capability of the encoder. Moreover, ClinicalBioBERT is trained by continuing the training of BioBERT on a small size of clinical data (about 2 million records), which may have led to the insufficient learning of clinical knowledge. The under-performance of ClinicalBioBERT does not necessarily mean that domain-specialized transformer models are inferior. Our experimental results also suggest that large pre-training data can boost the generalizability of models, while pre-training on small in-domain data may not benefit target tasks within the domain. Based on our findings, for social media text classification datasets, we recommend the use of RoBERTa-base, BERTweet or models pre-trained in similar fashion, and we do not recommend the use of ClinicalBioBERT, even for health-related social media tasks. A major limitation of our current work is that we only evaluated three pre-trained models, and, in the future, we will incorporate other similar models such as Twitter BERT (Dai et al., 2020) and BioBERT (Lee et al., 2019). We will also evaluate models using more metrics, as accuracy can be particularly misleading for imbalanced datasets.

5 Conclusion

We benchmarked the performances of three transformer-based pre-trained models on 25 social media text classification datasets. We found that RoBERTa-base and BERTweet perform similarly on most datasets, consistently outperforming ClinicalBioBERT, even for health-related tasks. Our experiments suggest that for social media-based classification tasks, it might be best to use

pre-trained on tweets (Twitter BERT) on 3 tweet classification tasks.

pre-trained models generated from large social media text. It might be possible to further improve the performance of BERTweet by incorporating data from multiple social networks.

References

Mohammed Ali Al-Garadi, Yuan-Chi Yang, Haitao Cai, Yucheng Ruan, Karen O'Connor, Graciela Gonzalez-Hernandez, Jeanmarie Perrone, and Abeed Sarker. 2020. Text Classification Models for the Automatic Detection of Nonmedical Prescription Medication Use from Social Media. *medRxiv*.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Shiladitya Bhattacharya, Siddharth Singh, Ritesh Kumar, Akanksha Bansal, Akash Bhagat, Yogesh Dawer, Bornini Lahiri, and Atul Kr. Ojha. 2020. Developing a Multilingual Annotated Corpus of Misogyny and Aggression. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 158–168, Marseille, France. European Language Resources Association (ELRA).

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2020. Cost-effective Selection of Pretraining Data: A Case Study of Pretraining BERT on Social Media. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1675–1681, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Debanjan Ghosh, Avijit Vajpayee, and Smaranda Muresan. 2020. A Report on the 2020 Sarcasm Detection Shared Task. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 1–11, Online. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, M. Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ArXiv*, abs/2007.15779.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining. *Bioinformatics*, 36(4):1234–1240.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv*, 1907(11692).

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 Task 1: Affect in Tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020a. BERTweet: A Pre-trained Language Model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

Dat Quoc Nguyen, Thanh Vu, Afshin Rahimi, Mai Hoang Dao, Linh The Nguyen, and Long Doan. 2020b. WNUT-2020 Task 2: Identification of Informative COVID-19 English Tweets. In *Proceedings of the 6th Workshop on Noisy User-generated Text*.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy. Association for Computational Linguistics.

Daniel Preoţiuc-Pietro, H Andrew Schwartz, Gregory Park, Johannes Eichstaedt, Margaret Kern, Lyle Ungar, and Elisabeth Shulman. 2016. Modelling Valence and Arousal in Facebook Posts. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 9–15.

Abeed Sarker, Mohammed Ali Al-Garadi, Yuan-Chi Yang, Sahithi Lakamana, Jie Lin, Sabrina Li, Angel Xie, Whitney Hogg-Bremer, Mylin Torres, Imon Banerjee, and Abeed Sarker. 2020. Automatic Breast Cancer Survivor Detection from Social Media for Studying Latent Factors Affecting Treatment Success. *medRxiv*.

Abeed Sarker, Maksim Belousov, Jasper Friedrichs, Kai Hakala, Svetlana Kiritchenko, Farrokh Mehryary, Sifei Han, Tung Tran, Anthony Rios, Ramakanth Kavuluru, Berry de Bruijn, Filip Ginter, Debanjan Mahata, Saif M Mohammad, Goran Nenadic, and Graciela Gonzalez-Hernandez. 2018.

Dataset	Description	Source
ADR Detection	Detect adverse reaction (ADR) mentioned from text	Sarker and Gonzalez (2015)
BreastCancer	Detect breast cancer patients based on their self-reports	Sarker et al. (2020)
PM Abuse	Identify prescription medication (PM) abuse on tweets	Ali Al-Garadi et al. (2020)[4]
SMM4H-17-task1	Detect adverse reaction (ADR) mentioned from text	Sarker et al. (2018)
SMM4H-17-task2	Identify medication consumption from medication-mentioning tweets	
WNUT-20-task2	Identify informative COVID-19 related tweets	Nguyen et al. (2020b)
OLID-1	Identify offensive language from tweets	Zampieri et al. (2019)
OLID-2		
OLID-3		
TRAC-1-1	Detect aggressive information in social media	Kumar et al. (2018)
TRAC-1-2		
TRAC-2-1	Detect aggressive language on social media text	Bhattacharya et al. (2020)
TRAC-2-2		
sarcasm-1	Binary emotion classification of sarcasm	Ghosh et al. (2020)
sarcasm-2		
CrowdFlower	Multiclass emotion classification	Web[5]
fb-arousal-1	Classify the level of arousal	Preoţiuc-Pietro et al. (2016)
fb-arousal-2		
fb-valence-1	Classify the level of valence	
fb-valence-2		
SemEval-18-A	Emotion intensity ordinal classification of anger	Mohammad et al. (2018)
SemEval-18-F	Emotion intensity ordinal classification of fear	
SemEval-18-J	Emotion intensity ordinal classification of joy	
SemEval-18-S	Emotion intensity ordinal classification sadness	
SemEval-18-V	Valence ordinal classification	

Table A.1: Data descriptions.

Data and Systems for Medication-Related Text Classification and Concept Normalization from Twitter: Insights from the Social Media Mining for Health (SMM4H)-2017 Shared Task. *Journal of the American Medical Informatics Association*, 25(10):1274–1283.

Abeed Sarker and Graciela Gonzalez. 2015. Portable Automatic Text Classification for Adverse Drug Reaction Detection via Multi-Corpus Training. *J. of Biomedical Informatics*, 53(C):196–207.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

A Appendix

A.1 Data Descriptions

Table A.1 provides a short description about the classification task focuses. The datasets that do not provide a split of train/dev/test sets are split into a training set and a test set using a 80/20 rate. For WNUT-20-task2, the results on the validation set was reported because the test set was not released.

A.2 Pooling Strategy Comparison

Table A.2 shows the results of taking `[CLS]` embeddings as document embeddings.

Dataset	RB		BT		CL	
	C	M	C	M	C	M
ADR Detection	91.7	91.4	90.4	**92.7**	90.8	90.4
BreastCancer	**94.1**	93.9	93.4	93.6	90.8	91.2
PM Abuse	81.1	81.4	81.9	**82.4**	77.4	77.4
SMM4H-17-task1	**93.6**	**93.6**	93.2	93.5	92.3	92.7
SMM4H-17-task2	78.9	78.4	79.1	**79.7**	74.3	75.0
WNUT-20-task2	**89.7**	89.1	88.3	88.3	85.8	86.5
OLID-1	**85.5**	85.1	84.7	85.2	83.4	83.5
OLID-2	89.2	89.4	**90.6**	90.0	89.2	89.0
OLID-3	68.5	69.5	**71.4**	70.0	67.8	66.4
TRAC-1-1	57.5	58.6	**59.2**	**59.2**	52.2	55.4
TRAC-1-2	58.6	58.8	**65.8**	**65.8**	57.4	58.0
TRAC2-1	**75.1**	72.8	63.3	73.3	66.3	63.9
TRAC2-2	85.4	85.8	83.9	85.5	**87.6**	87.3
CrowdFlower	39.8	39.9	35.0	**41.3**	38.8	38.8
fb-arousal-1	45.8	46.6	45.6	45.3	45.7	**46.8**
fb-arousal-2	54.6	**54.9**	52.9	54.8	52.4	54.1
fb-valence-1	59.5	60.2	60.5	**64.4**	52.9	54.5
fb-valence-2	**53.6**	52.8	52.6	52.6	44.9	45.9
sarcasm-1	66.3	67.3	**71.4**	69.5	64.8	64.6
sarcasm-2	73.2	73.3	**76.2**	76.1	68.0	68.2
SemEval-18-task-A	55.4	52.3	**60.8**	54.6	48.9	46.0
SemEval-18-task-F	49.4	47.7	43.4	**51.5**	45.1	45.3
SemEval-18-task-J	53.7	**54.9**	53.9	53.9	49.7	48.4
SemEval-18-task-S	68.2	**69.3**	64.2	67.4	65.9	65.3
SemEval-18-task-V	45.7	45.5	38.3	**46.6**	36.4	36.2

Table A.2: The accuracies of taking different pooling strategies on the test sets. `C`: [CLS] emebddings; `M`: mean word embeddings. The best results on each dataset are in boldface.

[5] https://projectreporter.nih.gov/project_info_description.cfm?aid=9577760

[5] https://data.world/crowdflower/sentiment-analysis-in-text

Pandemic Literature Search: Finding Information on COVID-19

Vincent Nguyen[1,2] **Maciej Rybinski**[1] **Sarvnaz Karimi**[1] **Zhenchang Xing**[2]
[1]CSIRO Data61, Sydney, Australia
[2]The Australian National University, Canberra, Australia
{firstname.lastname}@csiro.au
{zhenchang.xing}@anu.edu.au

Abstract

Finding information related to a pandemic of a novel disease raises new challenges for information seeking and retrieval, as the new information becomes available gradually. We investigate how to better rank information for pandemic information retrieval. We experiment with different ranking algorithms and propose a novel end-to-end method for neural retrieval, and demonstrate its effectiveness on the TREC COVID search.[1] This work could lead to a search system that aids scientists, clinicians, policymakers and others in finding reliable answers from the scientific literature.

1 Introduction

As COVID-19—an infectious disease caused by a coronavirus—led the world to a pandemic, a large number of scientific articles appeared in journals and other venues. In a span of five months, PubMed alone indexed over 60,000 articles matching coronavirus related search terms such as `SARS-CoV-2` or `COVID-19`. This volume of published material can be overwhelming. There is a need for effective search algorithms and question answering systems to find relevant information and answers. In response to this need, an international challenge—*TREC COVID Search* (Roberts et al., 2020; Voorhees et al., 2020)—was organised by several institutions, such as NIST and Allen Institute for AI, where research groups and tech companies developed systems that searched over scientific literature on coronavirus. Through an *iterative* setup organised in different rounds, participants are presented with several topics. The evaluations measure the effectiveness of these systems in finding the relevant articles containing answers to the questions in the topics.

We propose a method that improves the systems developed for the TREC-COVID challenge by adopting a novel hybrid neural end-to-end approach for ranking of search results. Our method combines a traditional inverted index and word-matching retrieval with a neural indexing component based on BERT architecture (Devlin et al., 2019). Our neural indexer leverages the Siamese network training framework (Reimers and Gurevych, 2019) fine-tuned on an auxiliary task (unrelated to literature retrieval) to produce universal sentence embeddings. This means that neural indexing can be performed offline for the entire document collection and does not need to be retrained on additional queries. This allows for incorporating the neural component for the entire retrieval process, contrasting with the typical multi-stage neural re-ranking approaches (Li et al., 2020; Zhang et al., 2020; Liu et al., 2017; Wang et al., 2011).

Our method is competitive with the top systems presented in TREC COVID [2]. It improves as corpus size increases despite not being trained on additional data which is a useful property in pandemic information retrieval.

2 Related Work

The use of neural networks in search has mostly been limited to reranking top results retrieved by a 'traditional' ranking mechanism, such as Okapi BM25 (Robertson et al., 1995). Only a portion of top results is rescored with a neural architecture (McDonald et al., 2018). Since the most successful neural reranking models depend on joint modelling of both documents and the query, rescoring the entire collection becomes costly. Moreover, the effectiveness gains achieved with neural reranking are debated (Yang et al., 2019) until recently (Lin, 2019).

Since late 2018, large neural models pre-trained on language modeling—specifically BERT (Devlin et al., 2019) which uses bi-directional transformer

[1]We release our code for the neural index in GitHub: https://git.io/JkZ7I

[2]https://git.io/JkZ7m Accessed: 10 Oct 2020

architecture—achieve state-of-the-art for several NLP tasks. The architecture is successfully applied to ad-hoc reranking (Nogueira and Cho, 2019; Akkalyoncu Yilmaz et al., 2019; Dai and Callan, 2019).

The existing applications of BERT in search share the limitation of being restricted to reranking because they rely on its next sentence prediction mechanism for a regression score. However, our approach builds on Reimers and Gurevych (2019), where a BERT architecture is trained to produce sentence embeddings. Leveraging these embeddings allows for a cost-efficient application of BERT to neural indexing.

Neural indexing is a less explored field. Whereas Zamani et al. (2018) leverages sparse neural representations for retrieval, Seo et al. (2019) uses sparse and dense representations for learning to rank. These methods rely on networks trained to produce representations directly for ranking documents. For our proposed method, we use universal embeddings[3] generated from transformer encoders trained on an auxiliary task of semantic similarity scoring or Natural Language Inference[4].

3 Dataset

Documents CORD-19 (The Covid-19 Open Research Dataset) (Wang et al., 2020) is a dataset of research articles on coronaviruses (COVID-19, SARS and MERS). It is compiled from three sources: PubMed Central (PMC), the WHO articles, and bioRxiv and medRxiv. Evaluations in subsequent stages (referred to as *rounds*) of TREC COVID Search task are performed on growing snapshots of CORD-19 dataset (Table 1). The collection grew to over 68,000 articles by mid-June 2020. The growth of CORD-19 continues with weekly updates (Roberts et al., 2020).

Topics As part of the TREC COVID search challenge, NIST provides a set of important COVID-related topics. Over five rounds, the topic set is augmented. Round 1 has 30 topics, with five new topics added per subsequent round. Each topic consists of three parts: query, question, and narrative (see Figure 1).

[3]The main property we are interested in for universal embeddings, is that pairs of embeddings can be compared directly via cosine similarity rather than indirectly comparing them through a task-specific network which requires additional training

[4]We do not directly use embeddings as a ranker as they are not trained for retrieval; instead, we use them in combination with a traditional inverted index.

Round	No. Documents	No. Judgments	No. Topics
1	51103	8691	30
2	59851	12037	35
3	128492	12993	40
4	157817	13312	45
5	191175	23373	50

Table 1: Statistics for each TREC-COVID round.

```
Topic 3
Query:      coronavirus immunity
Question:   will SARS-CoV2 infected
            people develop immunity?
            Is cross protection possible?
Narrative:  seeking studies of immunity
            developed due to infection with
            SARS-CoV2 or cross protection
            gained due to infection with
            other coronavirus types
```

Figure 1: A sample topic from the TREC COVID.

Relevance Judgements and Evaluation TREC organises manual judgements per each round of the shared task, using a pooling method over a sample of the submitted runs (Voorhees et al., 2020). Given a topic, a document is judged as: irrelevant (0), partially relevant (1), and relevant (2). As judgements are manually annotated by biomedical experts, only a subset of runs submitted to the track are judged.

The evaluation procedure in each of the subsequent rounds discards (topic, document) pairs included judged in previous rounds. We use this procedure (referred to as *residual scoring*) when comparing against the top-performing runs in the competitive.

In additional experiments, we use *cumulative scoring*, which means evaluating topics for round 2 using human judgments for rounds 1 and 2. Topics of round 3 are evaluated using judgments of rounds 1–3, and so on. Using cumulative scoring allows us to use a larger proportion of judged documents for the topic sets corresponding to subsequent rounds.

Metrics Four precision focused metrics are used to evaluate the rankings: NDCG (Järvelin and Kekäläinen, 2002) at rank 10 (NDCG@10), precision at rank 10 (P@10), mean average precision (MAP) and recall-precision (R-prec). BPref takes into account the noisy and incomplete judgements.

4 Methods

Neural Index Retrieval (NIR) We built a hybrid neural index by appending neural representation vectors to document representations of a traditional inverted index. The neural representations are created using an average over individual representations of sentences (bag-of-sentences) from a BERT-based universal sentence encoder for the title, abstract and full-text facets. Sentence representations are created by averaging token-level representations produced by the encoder (average pooling strategy outlined in Reimers and Gurevych (2019)). We investigate a selection of models derived from applying the training of the Sentence Transformer (Reimers and Gurevych, 2019), a Siamese network built to enable cosine comparability between transformer sentence embeddings, and the biomedically-themed BERT-based pre-trained models, such as BioBERT (Lee et al., 2019). To obtain individual sentences, we use a neural sentence segmentation model, *ScispaCy* (Neumann et al., 2019).

For retrieval, we propose a hybrid approach. We score (topic, document) pairs by combining: (1) Okapi BM25 scores for all pairs of topic fields and document facets; and, (2) cosine similarities calculated for neural representations of all pairs of topic fields (calculated *ad hoc*) and document facets stored in the index[5]. The final score adds a log-normalised sum of BM25 scores to the sum of neural scores. Formally, the relevance score ψ for ith topic T_i and document $d \in D$ is

$$\psi(T_i, d) = \log_z(\sum^{t \in T_i} \sum^{f \in d} BM25(t, f)) + \sum^{t \in T_i} \sum^{f \in d} cos(v(t), v(f)), \quad (1)$$

where z is a hyper-parameter, $t \in T_i$ represents fields of the topic (i.e., query, narrative and question), $f \in d$ represents facets of the document (i.e., abstract, title, body), BM25 denotes the BM25 scoring function, $v(t)$ is the neural representation of the topic field, $v(f)$ denotes the neural representation of the document facet, and cos is cosine similarity. The hyper-parameter z is solved for each topic with the formula:

$$z = \sqrt[R_{cos}]{max(BM25(t, f))} \quad (2)$$

where R_{cos} is the upper range of the summed cosine function:

$$R_{cos} = max(\sum^{t \in T_i} \sum^{f \in d} cos(v(t), v(t))) \quad (3)$$

The z hyper-parameter normalizes the BM25 score such that its range will be the same as the range of the summed cosine similarity score, R_{cos}. This is to ensure both components, neural and BM25, have equal contribution to the final score.

We also filter by date. The documents created before December 31st 2019 (the first reported COVID-19 case) are removed.

Sentence Embedding Models We compare four different embedding models. We choose our models based on differences in pre-training corpora (PubMed vs. different COVID-specific corpora) and Siamese fine-tuning task (NLI, Natural Language Inference), and STS (Semantic Textual Similarity). We evaluate BioBERT-NLI and BioBERT-STS (pre-trained on PubMed corpus, before COVID), CovidBERT-NLI (pre-trained on a small subset of CORD corpus), and ClinicalCovidBERT-NLI (pre-trained on a larger subset of the CORD corpus)[6].

As a baseline, we use our method with a BioBERT model fine-tuned on an *ad hoc* retrieval task on MS Marco dataset (Nguyen et al., 2016). BioBERT-msmarco is not a universal sentence encoder, and its inclusion is to provide perspective on the significance of using the Siamese fine-tuning in our neural indexing approach. Additionally, we include BM25 as a baseline.

BM25 and top-run baselines For each evaluation round, we report an unmodified BM25 (no neural index) baseline together with a top automatic run (per official *leaderboard*) from the TREC evaluations. Note that the best run baseline does not refer to one specific system, but the best performing run for each round of the evaluation.

5 Experimental Results

We present: (1) a comparison of retrieval effectiveness of our method with different embedding models using cumulative scoring on rounds 1–3 (Table 2); (2) a comparison of our most effective system to the BM25 baseline and best runs from the official shared task evaluations using residual

[5] We emphasise that our model is not a re-ranking model but a ranker model as it scores the entire collection during retrieval, rather than re-ranking a retrieved list.

[6] A directory to the models: https://git.io/JTfz2 Accessed: 10 Oct 2020

Model	Round 1					Round 2					Round 3				
	NDCG@10	P@10	bpref	MAP	R-prec	NDCG@10	P@10	bpref	MAP	R-prec	NDCG@10	P@10	bpref	MAP	R-prec
BM25 (baseline)	0.624	0.650	0.409	0.258	**0.316**	**0.666**	0.694	0.380	0.240	0.304	0.717	0.762	0.412	0.235	0.313
BioBERT-NLI	0.614	0.597	0.384	0.219	0.279	0.608	0.671	0.374	0.219	0.291	0.726†	0.772	0.410	0.237	0.311
Covid-NLI	0.582	0.597	0.409	0.249	0.309	0.522	0.597	0.347	0.193	0.274	0.736	0.780	0.413	0.239	0.316
ClinicalCovid-NLI	**0.641**†	**0.663**	**0.408**	**0.258**	0.315	0.650	**0.710**	**0.397**	**0.264**	**0.320**	**0.739**	**0.780**	**0.420**‡	**0.252**†	**0.327**
BioBERT 1.1 STS	0.612	**0.633**	**0.408**	0.246	0.302	0.613	0.663	0.396	0.251	0.314	0.722	0.762	0.398	0.228	0.302
BioBERT msmarco	0.528	0.530	0.366	0.197	0.255	0.593	0.666	0.372	0.232	0.303	0.691	0.743	0.389	0.218	0.296

Table 2: Results for our runs for Round 1–3. Best run is as reported by organisers per that round. † denotes statistical significance at 95% and ‡ at 99% over the baseline.

Model	Round 1					Round 2					Round 3					Round 4					Round 5				
	NDCG@10	P@10	bpref	MAP	R-prec	NDCG@10	P@10	bpref	MAP	R-prec	NDCG@10	P@10	bpref	MAP	R-prec	NDCG@10	P@10	bpref	MAP	R-prec	NDCG@10	P@10	bpref	MAP	R-prec
BM25	0.614	0.633	0.407	0.257	0.314	0.607	0.634	0.398	0.222	0.278	0.569	0.618	0.361	0.174	0.243	0.666	0.724	0.461	0.247	0.300	0.632	0.672	0.378	0.190	0.267
ClinicalCovid-NLI	**0.661**	0.663	0.422	0.258	0.322	**0.626**	0.646	0.410	0.228	0.289	0.600	0.647	0.384†	0.193†	0.262	0.685	0.716	0.492‡	0.262	0.314	0.709‡	0.770‡	0.428	0.230‡	0.298‡
Best Automatic Run	0.608	**0.700**	**0.483**	**0.313**	**0.355**	0.625	**0.657**	**0.457**	**0.284**	**0.325**	**0.671**	**0.748**	**0.560**	**0.305**	**0.347**	**0.791**	**0.818**	**0.557**	**0.311**	**0.342**	**0.727**	**0.782**	**0.550**	**0.320**	**0.378**

Table 3: Our proposed method with comparison to a BM25 baseline and the top automatic run for that round. All evaluation is performed with residual document scoring

scoring on rounds 1–5 (Table 3); and, (3) an ablation test of our most effective system on round 5 topics using cumulative scoring (Table 4).

Choice of the embedding model Table 2 provides insights into the selection of the sentence embedder: (1) the importance of domain-adaptive pre-training for neural re-ranking, that is using a model pre-trained on a task-specific corpus. We believe it is especially important in our setup, as there is no other task-specific training involved at any stage. Unsurprisingly, using a larger domain-specific corpus in pre-training yields better results; (2) there is no apparent difference between NLI and STS fine-tuning. Notably, BioBERT-msmarco performs worse than other evaluated models and the baseline, showing the importance of adapting BERT to act as a universal sentence encoder at the fine-tuning stage.

Ablations Table 4 confirms that the combination of BM25 with the neural indexing yields best overall results as removing either component leads to a significant loss in performance. Removal of facets makes no significant differences. Removal of the date filter significantly degrades NDCG@10.

Comparisons with best runs Aside from rounds 3 and 4, our models remain competitive with the top run. Our model scored higher NDCG@10 for rounds 1 and 2 over the baseline automatic runs. Most of the top runs used neural re-rankers which have been specifically trained on related tasks such as med-marco (MacAvaney et al., 2020).

Where does the model succeed or fail? Our model consistently outperforms the BM25 baseline (Table 3).

Model	P@10	NDCG@10
NIR	0.852	0.796
no neural	0.744†	0.808†
no BM25	0.668‡	0.706‡
no title	0.848	0.784
no abstract	0.848	0.785
no fulltext	**0.856**	**0.799**
no date filter	0.834	0.775†

Table 4: Ablation studies for our proposed method where document facets, query facets and other aspects of the model are removed.

The model can retrieve documents undiscovered by the BM25 component or a pipeline model which uses word-overlap scoring in its initial retrieval. It computes scores over the entire collection as a hybrid inverted index which leads to an average increase of in 6% R-prec values (Table 3) over the BM25 baseline. The improvement in early recall is also a desirable feature if we were to pair our model with a task-specific neural re-ranker.

We expect that the top ranked documents are scored highly by both components, however, we found that our model placed an irrelevant document at the rank one for Topic 3. This document was scored highly by BM25 but much lower in the neural/cosine component. It saturated the scoring function as it repeated many of the keywords in the query, however, the semantic content of the text was irrelevant to the query itself as it discussed "coronavirus crossing continents" rather than "coronavirus cross protection".

On the other hand, for topic 1, "coronavirus origins", we found that the neural index overcame semantic mismatches of the BM25 scoring. In the dataset, most documents are related to coron-

avirus, the word "origin" contributes more to the final score and BM25 retrieved an irrelevant document at rank three which is a document that discusses origins of a different virus. However, when using the neural scorer, this document is placed at rank 42.

From Table 2 and 3, although our model is not trained on any additional data, it improves in ranking as the corpus size increases. This is a useful property in pandemic information retrieval as the model does not need to be continually retrained, and each document is embedded once.

6 Conclusions

We propose a novel neural ranking approach (NIR) for pandemic information retrieval. Experimenting with the TREC COVID search challenge, we show that our method is competitive compared to other automatic systems. We show that a neural scoring is beneficial in alleviating some of the shortcomings of the keyword-based retrieval. Empirically, our model shows improvements with time in a pandemic scenario without additional training data. A balanced scoring function combines the strengths of the inverted and neural indices. A neural index explicitly trained for ranking would be a suitable avenue for future research.

Acknowledgements

Vincent is supported by the Australian Research Training Program and the CSIRO Research Office Postgraduate Scholarship. This work is funded by the CSIRO Precision Health Future Science Platform.

References

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *EMNLP*, pages 3490–3496, Hong Kong, China.

Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *SIGIR*, pages 985–988, Paris, France.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, pages 4171–4186, Minneapolis, MN.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: passage representation aggregation for document reranking. *arXiv:2008.09093*.

Jimmy Lin. 2019. Neural hype, justified! A recantation. *ACM SIGIR Forum*, 53.

Shichen Liu, Fei Xiao, Wenwu Ou, and Luo Si. 2017. Cascade ranking for operational e-commerce search. In *SIGKDD*, page 1557–1565, New York, NY, USA.

Sean MacAvaney, Arman Cohan, and Nazli Goharian. 2020. SLEDGE: A simple yet effective baseline for COVID-19 scientific knowledge search. *arXiv:2005.02365*.

Ryan McDonald, George Brokos, and Ion Androutsopoulos. 2018. Deep Relevance Ranking Using Enhanced Document-Query Interactions. In *EMNLP*, pages 1849–1860, Brussels, Belgium.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *BioNLP*, pages 319–327, Florence, Italy.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *NIPS Cognitive Computations Workshop*, volume 1773, Barcelona, Spain.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv:1901.04085*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP*, pages 3982–3992, Hong Kong, China.

Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William Hersh. 2020. TREC-COVID: Rationale and structure of an information retrieval shared task for COVID-19. *J. Am. Med. Inform. Assoc*.

Stephen Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1995. Okapi at TREC-3. In *TREC*, Gaithersburg, MD, US.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *ACL*, pages 4430–4441, Florence, Italy.

Ellen Voorhees, Alam Tasmeer, Demner-Fushman Dina, Hersh William, and Kyle Lo. 2020. TREC-COVID: Constructing a pandemic information retrieval test collection. *ACM SIGIR Forum*, 54:1–12.

Lidan Wang, Jimmy Lin, and Donald Metzler. 2011. A cascade ranking model for efficient ranked retrieval. In *SIGIR*, page 105–114, Beijing, China.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, Paul Mooney, Dewey Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The Covid-19 Open Research Dataset. In *ACL NLP-COVID Workshop*, Online.

Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. 2019. Critically examining the "neural hype": Weak baselines and the additivity of effectiveness gains from neural ranking models. In *SIGIR*, pages 1129–1132, Paris, France.

Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *CIKM*, page 497–506, Torino, Italy.

Edwin Zhang, Nikhil Gupta, Raphael Tang, Xiao Han, Ronak Pradeep, Kuang Lu, Yue Zhang, Rodrigo Nogueira, Kyunghyun Cho, Hui Fang, and Jimmy Lin. 2020. Covidex: Neural ranking models and keyword search infrastructure for the COVID-19 open research dataset. *arXiv:2007.0784*.

Information Extraction from Legal Documents: A Study in the Context of Common Law Court Judgements

Meladel Mistica[♦]* **Geordie Z. Zhang**[♥]* **Hui Chia**[♣]*
Kabir Manandhar Shrestha[♥] **Rohit Kumar Gupta**[♥] **Saket Khandelwal**[♥]
Jeannie Marie Paterson[♣] **Timothy Baldwin**[♦] **Daniel Beck**[♦]
[♦] School of Computing and Information Systems
[♥] Melbourne Data Analytics Platform [♣] Melbourne Law School
The University of Melbourne, Australia
{misticam, geordie.zhang, chia.h}@unimelb.edu.au
{kmanandharsh, rohitkumarg, saketk}@student.unimelb.edu.au
{jeanniep, tbaldwin, d.beck}@unimelb.edu.au

Abstract

'Common Law' judicial systems follow the doctrine of precedent, which means the legal principles articulated in court judgements are binding in subsequent cases in lower courts. For this reason, lawyers must search prior judgements for the legal principles that are relevant to their case. The difficulty for those within the legal profession is that the information that they are looking for may be contained within a few paragraphs or sentences, but those few paragraphs may be buried within a hundred-page document. In this study, we create a schema based on the relevant information that legal professionals seek within judgements and perform text classification based on it, with the aim of not only assisting lawyers in researching cases, but eventually enabling large-scale analysis of legal judgements to find trends in court outcomes over time.

1 Introduction

The law is reason free from passion[1] — but you'll have to dig through hundreds of pages to find it.

In common law countries such as Australia, a core legal principle is the doctrine of precedent — every court judgement contains legal rulings that are binding upon subsequent cases in lower courts, though how legal rulings apply in subsequent cases is dependent on the facts of the case. When preparing to give a legal opinion or argue a case, lawyers spend many long hours reading lengthy judgements to identify therein the precedents that are salient to the case at hand. This time-consuming manual process has formed a barrier to large-scale analysis of legal judgements. Even though thousands of court judgements are published in Australia every year,[2] lawyers are only able to analyse small numbers of judgements, potentially missing broader trends hidden in the vast numbers of judgements that are published by the courts.

There is a growing body of research at the intersection of Law and Natural Language Processing, including prediction of court opinion about a case (Chalkidis et al., 2019a; Aletras et al., 2016), classification of legal text by legal topics or issues (Soh et al., 2019; Chalkidis et al., 2019b), and legal entity recognition (Cardellino et al., 2017). However, our ultimate goal is to assist lawyers in identifying sections of judgements relevant to their case at hand, as well as bulk analysis of cases to identify relationships between factual patterns and decision outcomes. For this reason, we model our initial study on the sentence-by-sentence identification of argumentation zones within academic and scientific texts (Teufel et al., 2009; Guo et al., 2010). However, these zoning papers do not account for the complex document structure of legal judgements, which have the potential to be structured as multiple sub-documents within the one court decision (see Section 3).

The overall goal of the project is to automate the extraction of information from legal judgements, to assist lawyers to more easily and quickly identify the type of information that they are looking for from a large number of judgements. The project also aims to enable the large-scale analysis of judgements by legal researchers in order to identify trends or patterns that may be occurring within judgments, for example identifying patterns

* Meladel Mistica, Geordie Z. Zhang, and Hui Chia contributed equally to this paper.

[1] Aristotle, Politica (Politics By Aristotle), written 350 B.C.E, translated by Benjamin Jowett

[2] For example, the Federal Court of Australia alone publishes around 1700–2500 judgements per year.

of facts that lead to particular results. This kind of analysis is relevant in predicting the outcome of complex cases and may also inform law reform. This part of the study reports on the initial phase of experimenting with the granularity of the annotation labels in developing our schema, as well as our initial experiments in automatically identifying these labels.

2 Background

Legal research as a broad term can include any form of research that is undertaken for the purpose of advancing legal advice, litigation or law reform, and can include research activities such as community surveys, comparative studies of legislation and the study of court judgments.

This project focuses solely on the activity of studying court judgments, as it is a crucial component of legal research in common law countries. For lawyers and legal researchers, court judgments are a key source of data for the purpose of legal research, though legal research in general can encompass other sources of data, such as legislation, international treaties, government reports etc.

When lawyers or legal researchers read a court judgment, what they are looking for is observations, opinions or decisions that the judge has made about how the law should be interpreted and applied in the particular context of the case before it. For example, what are the rules to resolve conflict between competing values, or what are the rules for resolving ambiguities of the meaning of a word in legislation? These observations, opinions and decisions by judges can be conceptualised as "law data" – data that legal researchers collect in order to understand how laws are being applied by courts to specific factual patterns and to predict how it may be applied in future scenarios.

Collecting data about how laws are interpreted is important at both the individual and the societal level. At the individual level, much of a lawyer's work is advising clients on what they need to do to comply with the law. Lawyers will research past court judgments to collect data about how the law has been interpreted in similar factual situations, in order to make an informed opinion about how the law is likely to be applied to the case at hand. At the societal level, legal researchers in academia, regulatory agencies and government collect data on how laws are being interpreted and applied to specific facts, in order to assess whether laws are delivering the desired social outcomes.

The field of legal research has conventionally relied mostly on qualitative data, and if there is quantitative data it is usually at a small scale. The reason for this is because "law data" is expressed in court judgments that are generally very long and complex free-form text. The only method for collecting "law data" has been through the manual reading of legal judgments by people with legal expertise. This is a very time-consuming process and therefore legal research has generally had to rely on small quantities of data.

The contribution that NLP can make to the legal field is to enable the automatic extraction of "law data" from court judgements, to increase the number of court judgments that legal researchers can analyse. The challenge for this project has been the novelty of the task of extracting complex data from court judgments. There is no established schema for extracting information from court judgments. The schema proposed in this study is the result of a multi-disciplinary approach to merging the categories of data that are useful to legal researchers and lawyers, with the categories of information that can be accurately labelled using text classification.

3 Corpus Development

We developed our initial proof-of-concept corpus from court judgements from the High Court of Australia,[3] which is the highest court in the Australian judicial system hierarchy. A court case may be decided by a single judge or a group of judges. In the case of a single judge, the court judgement is single-authored with one voice. When there are multiple judges, they can write a single judgement as a group, particularly if they are in agreement, or they can give separate reasons. In the latter case, the court judgement will then consist of multiple sets of reasons, structured as sub-components from the different judges, which together make up the entire judgement for that court case.

To legal domain experts, there are general patterns or sequences by which different types of information tend to appear within a judgement. However there is a high degree of variation between court judgements according to the writing style of the judge. For instance, one common document pattern begins with the explanation of the facts of the case, followed by the reasoning on how the rele-

[3] https://www.hcourt.gov.au/publications/judgements

LABEL	DESCRIPTION
FACT	Specific facts of that case, e.g. *The applicant entered Australia as an unauthorised maritime arrival on 5 September 2011.*
REASONING	Legal principles considered, e.g. *The question that arises is whether the Tribunal failed to consider that the applicant faced a real probability of irreparable harm.*
CONCLUSION	Outcome of the case, e.g. *The Tribunal committed a jurisdictional error, the appeal should be allowed.*

Figure 1: Description of the Label Set

vant legal principles were applied, and then ending with their conclusion. But this is not always the case. Some judges will state their conclusions at the beginning, and then provide a detailed examination of the facts and legal reasoning. Where there are multiple sets of reasons within a single judgement, each set of reasons will have its own structure particular to that judge's writing style.

We limit our corpus to immigration law cases, and randomly selected 55 of these High Court judgements. These 55 documents contain over 9.5K sentences in total. Each of them was annotated at the sentence level with either FACT, REASONING or CONCLUSION, which capture different aspects of the case as shown in Figure 1. In this initial corpus, REASONING made up half of the labelled sentences. Of the remaining sentences, three quarters were labelled FACT, and one quarter CONCLUSION. The FACT and CONCLUSION segments of the case are usually what lawyers are most interested in. These portions of the document (judgement) contain unique details pertaining to the case, while the REASONING category is a combination of original insights of this case and a recapitulation of previous relevant judgements.

Annotation For the annotations, we had 1 primary annotator (ANNOTATOR A), a qualified lawyer and legal researcher, who marked up all of the sampled High Court judgements. ANNOTATOR A had a label distribution of FACT: 38%, REASONING: 50%, and CONCLUSION: 12%. We also had 2 secondary annotators (ANNOTATORS B and C): the first is a practising immigration lawyer, and the second has some legal training, but is not a fully qualified lawyer. We randomly selected 3 documents (judgements) for the secondary annotators to mark up. This made up 5% of the number of sentences of the whole corpus. Of those sentences, there were no three-way disagreements between the annotators. The Cohen's kappa (κ) between all three annotators shows very good 2-way agreement between all pairs of annotators. The inter-annotator agreement between A–B and B–C were 0.70, and between A–C was 0.73. A large majority of the 2-way disagreements involved REASONING, with 81.5% of the disagreements being REASONING-vs-FACT and REASONING-vs-CONCLUSION, split roughly 50:50.[4]

4 Experiments

In order to assess the feasibility of using our corpus in a supervised setting, we perform experiments using a range of different models for sentence-level classification. The goal is to have a reasonable understanding of how difficult the task is, both in terms of our initial schema and training data size.

Data Processing Although the task is modelled at the sentence level, the corpus was split at the document-level for training, validation, and testing. This set-up emulates the real-world setting, where new documents are classified as a whole. We use a 80%:10%:10% split for training, development and testing (corresponding to 44:5:6 documents and 3000:1200:800 sentences, respectively). Since there is a smaller number of CONCLUSION sentences in court judgements, we perform undersampling over the training data only, by randomly deleting samples from the other majority classes to balance the number of training instances across the three labels. Note that this was performed for the training set only, and the development and testing sets were left untouched.

Methods As two baselines, we use: (1) a majority-class classifier, based on the training data;

[4]We note that the dataset will not be made publicly available because the project team does not have the right to publish this data. Whilst court judgments are in the public domain, there are copyright restrictions on republication. Republication of court judgments in an altered form, which our labelled dataset would be, is not allowed.

Model	P	R	Macro F1	Micro F1
RoBERTa	.64	.67	.65	.71
BERT	.64	.70	.65	.70
XLNet	.65	.70	.66	.72
MajorityClass	.20	.33	.25	.59
NBSVM	.55	.56	.55	.63

Table 1: Initial Performance Evaluation

and (2) the NBSVM model proposed by Wang and Manning (2012), which combines a naive Bayes model with a support vector machine, using a bag-of-words text representation. We compare this with a set of pre-trained language models, namely BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019). We employ similar structures for these models: 12 layers of transformer blocks, a hidden layer size of 768d, and 12 attention heads. All models are trained by adding a single hidden layer with softmax output.[5]

Initial Results We evaluate our models using Precision, Recall, and Macro-averaged and Micro-averaged F1, showing the results in Table 1. The NBSVM model outperforms the majority class baseline by 0.30 in Macro F1. Using a pre-trained model further improves the performance, with XLNet increasing Macro F1 by 0.11 over the NBSVM baseline, and achieving the best results. While this is expected, since these models have been pre-trained over large amounts of textual data, it is still remarkable given how domain-specific court judgements are.

Incorporating Context While our initial results are promising, at 0.66 Macro F1 they still result in many errors. This undermines the potential of our approach to be deployed in real-world scenarios. In the remaining experiments, we explore a few approaches to improve performance, focusing on XLNet since it was our best model in the initial experiments.

One hypothesis is that the label of a sentence is affected by its context in the document. This is directly reflected in the annotation procedure, since annotators have access to the full document when labelling sentences. In order to test this hypothesis, we prepend each sentence with its two previous sentences in the document, and feed the sequence of three sentences into XLNet as input.

[5]We refer the reader to the original papers from each model for details of the architecture and model pre-training.

Model	Class	P	R	F1
XLNet	CONCLUSION	.42	.71	.53
	FACT	.72	.62	.67
	REASONING	.81	.77	.79
$\text{XLNet}_{\text{context}}$	CONCLUSION	.58	.80	.67
	FACT	.85	.83	.84
	REASONING	.82	.74	.78

Table 2: Results for XLNet without & with Sentential Context (Prepending the Previous Two Sentences)

Model	Class	P	R	F1
$\text{XLNet}_{\text{context}}$	CONCLUSION	.71	.57	.63
	FACT	.85	.85	.85
	REASONING	.83	.87	.85

Table 3: Results for $\text{XLNet}_{\text{context}}$ with Sentential Context but without Undersampling

We show the results of this approach in Table 2, comparing with the XLNet model used in the initial experiments without sentential context. We also break down the results across the three individual classes, to get a better understanding of any differences in performance. Overall, adding context greatly improves the performance in detecting FACT and CONCLUSION sentences, reaching an overall Macro F1 of 0.76 and Micro F1 of 0.79, a 0.10 and 0.07 improvement over the single sentence model, respectively. Interestingly, adding context does not seem to affect REASONING sentences much, with a small decrease in Recall. This could be evidence that REASONING sentences can be detected only by local content within the sentence, without necessarily requiring extra-sentential context.

Effect of Undersampling We also investigated the impact of undersampling the training data. Our motivation for undersampling is the unbalanced nature of the dataset, where around half of the sentences are labelled as REASONING. This is an issue since, as explained in Section 3, legal experts are mostly interested in FACT and CONCLUSION sentences.

In Table 3 we present the results for $\text{XLNet}_{\text{context}}$ without undersampling, to compare against the

original results in (the bottom half of) Table 2 with undersampling. The results show a drop in recall for CONCLUSION, which was expected, while improving the recall for REASONING. FACT, however, was largely unaffected. Note that recall is particularly critical in our use case, in highlighting potential FACT and CONCLUSION sentences to our legal expert.

5 Discussion and Future Work

In this paper, we have presented the preliminary investigations of our interdisciplinary collaboration. The main focus was to scope out the areas in which NLP can assist in the task of interpreting legal judgments — a task that every lawyer must do in researching a case. The main contribution of this paper is developing and testing the annotation schema. In future work, we aim to extract trends over time for a given aspect of the annotation, e.g. how the presentation REASONING changes over time as new cases are judged with each new CONCLUSION. Given that Australia has a common law system, these judgements in effect shape the interpretation and understanding of the law and set a precedence for subsequent cases.

The results of the sentence-level text classification are promising despite the inherent confusability within the REASONING class: even professional lawyers with years of training can disagree in ascertaining whether a sentence is indeed a REASONING rather than a CONCLUSION or in some cases a REASONING or a FACT sentence, as there can be elements of either within a REASONING sentence. Although the results do show promise, in future work, we intend to experiment with the annotation schema to explore more detailed sub-categories under REASONING. This will assist us in identifying more targeted zones within the judgements, which may better assist in legal information extraction tasks, and in better characterising the structure of these legal documents.

From an application perspective, we plan to test the newly released LegalBERT (Chalkidis et al., 2020) and compare this to our adaptation of a domain-specific BERT and XLNet for legal texts. We note that LegalBERT was pre-trained on a variety of legal texts that are different from the legal texts in our database, which consisted solely of Australian court judgments. The data used to pretrain LegalBERT included legislation and contracts, which are different to court judgments in terms of structure and content. Also, the data used to pretrain LegalBERT was from multiple legal jurisdictions, being the United States, United Kingdom and Europe, with each jurisdiction having unique nuances to the language used in its legal texts. Given these differences between our data and the training data of LegalBERT, it remains an open question as to whether LegalBERT would have any advantage over BERT, and whether a custom-tuned BERT for our purposes may be more advantageous.

Acknowledgements

This work was supported in part by the Melbourne Data Analytics Platform. We are grateful for the discussions and suggestions from Priyanka Pillai, Emily Fitzgerald, Daniel Russo-Batterham, Kim Doyle, and Andrew Turpin in the shaping of this project. The authors would also like to thank the reviewers for their constructive feedback.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93.

Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. 2017. Legal NERC with ontologies, Wikipedia and curriculum learning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 254–259, Valencia, Spain. Association for Computational Linguistics.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019a. Neural legal judgment prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2019b. Extreme multi-label legal text classification: A case study in EU legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 78–87, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yufan Guo, Anna Korhonen, Maria Liakata, Ilona Silins, Lin Sun, and Ulla Stenius. 2010. Identifying the information structure of scientific abstracts: An investigation of three different schemes. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 99–107, Uppsala, Sweden. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv*, abs/1907.11692.

Jerrold Soh, How Khang Lim, and Ian Ernst Chai. 2019. Legal area classification: A comparative study of text classifiers on Singapore Supreme Court judgments. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 67–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Simone Teufel, Advaith Siddharthan, and Colin Batchelor. 2009. Towards domain-independent argumentative zoning: Evidence from chemistry and computational linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1493–1502, Singapore. Association for Computational Linguistics.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Convolutional and Recurrent Neural Networks for Spoken Emotion Recognition

Aaron Keesing
School of Computer Science
University of Auckland
New Zealand
akee511@aucklanduni.ac.nz

Ian Watson
School of Computer Science
University of Auckland
New Zealand
ian@cs.auckland.ac.nz

Michael Witbrock
School of Computer Science
University of Auckland
New Zealand
m.witbrock@auckland.ac.nz

Abstract

We test four models proposed in the speech emotion recognition (SER) literature on 15 public and academic licensed datasets in speaker-independent cross-validation. Results indicate differences in the performance of the models which is partly dependent on the dataset and features used. We also show that a standard utterance-level feature set still performs competitively with neural models on some datasets. This work serves as a starting point for future model comparisons, in addition to open-sourcing the testing code.

1 Introduction

Speech emotion recognition (SER) is the analysis of speech to predict the emotional state of the speaker, for which there are many current and potential applications (Peter and Beale, 2008; Koolagudi and Rao, 2012). As speech-enabled devices become more prevalent, the need for reliable and robust SER increases, and also the need for comparability of results on common datasets. While there has been a large amount of research in this field, a lot of results come from testing only on one or two datasets, which may or may not be publicly available. Additionally, different methodologies are often used, reducing direct comparability of results. Given the wide variety of neural architectures and testing methodologies, there is need for a common testing framework to help comparisons.

This study aims to test some SER models proposed in the literature on a discrete emotion classification task, and promote reproducibility of results by using public and academic licensed datasets. In addition, the code is publicly hosted on GitHub[1] under an open source license, so that our results may be verified and built upon. Our work has two main benefits. First, it serves as a baseline reference for future research that uses datasets present in this study. Second, it allows for comparisons between datasets to see which of their properties may influence classification performance of different models.

The paper is structured as follows. In Section 2 related work is given, and in Section 3 we list the datasets used in this study. The tested methods are outlined in Section 4, and the results given in Section 5. We briefly discuss these results in Section 6 and a conclude in Section 7.

2 Related Work

There has been some previous work in comparing SER techniques on a number of datasets. In Schuller et al. (2009a), Schuller et al. compare a hidden Markov model/Gaussian mixture model (HMM/GMM) and a SVM classifier for emotion class, arousal and valence prediction on nine datasets. For HMM/GMM, 12 MFCC, log-frame-energy, speed and acceleration features, are extracted per frame. For SVM, 6552 features are extracted based on 39 statistical functionals of 56 low-level descriptors (LLDs). Testing was done in a leave-one-speaker-out (LOSO) or leave-one-speaker-group-out (LOSGO) cross-validation setup. The only three datasets in common with the present study are EMO-DB, eNTERFACE and SmartKom, for which unweighted average recall (UAR) of 84.6%, 72.5%, and 23.5% were achieved, respectively. We use a similar methodology in the present paper.

The Schuller et al. work is expanded in Stuhlsatz et al. (2011), where multi-layer stacks of restricted Boltzmann machines (RBMs) are pre-trained in an unsupervised manner, then fine-tuned using backpropagation as a feed-forward neural network. The same datasets and configurations are used

[1] https://github.com/Broad-AI-Lab/emotion

	Emotion									
Dataset	ang.	hap.	sad.	fear	sur.	dis.	neu.	oth.	unk.	Total
CaFE	144	144	144	144	144	144	72			936
CREMA-D	1271	1271	1270	1271		1271	1087			7441
DEMoS	246	167	422	177	203	140		209		1564
EMO-DB	127	71	62	69		46	79	81		535
EmoFilm	232	240	254	221		168				1115
eNTERFACE	215	212	215	215	215	215				1287
IEMOCAP	1103	1636	1084				1708			5531
JL-corpus	240	240	240				240	240		1200
MSP-IMPROV	792	2644	885				3477			7798
Portuguese	63	46	59	41	64	35	60			368
RAVDESS	192	192	192	192	192	192	96	192		1440
SAVEE	60	60	60	60	60	60	120			480
ShEMO	1059	201	449		225		1028			2962
SmartKom	99	118	54		9		1786	183	46	2295
TESS	400	400	400	400	400	400	400			2800

Table 1: Dataset emotion distribution. The number of clips in each of the 'big six' emotions along with neutral and other, is given, as well as the total number of clips in each dataset. *oth.* = other (dataset specific); *unk.* = unknown

as in Schuller et al. (2009a), but the all-class emotion classification results are better on only some of the datasets. In particular, GerDA performs slightly better on average for SmartKom, but slightly worse for EMO-DB and eNTERFACE. In the current work, we compare many more methods on many more datasets; we also include more recent datasets.

3 Datasets

Fifteen datasets are used in this study, some of which are open datasets, while others require a signed EULA to access. All of the datasets have a set of categorical emotional labels. A question arises when using acted datasets with additional annotations, such as CREMA-D, as to whether to use the actor's intended emotion as 'ground truth' for training a classifier or instead use a consensus of annotators with majority vote. For MSP-IMPROV and IEMOCAP, the label assigned by annotators is used, consistent with previous work. For CREMA-D we have opted to use the actors intended emotion, rather than any annotator assigned labels. A table describing the emotion distribution in each dataset is given in Table 1.

3.1 Open Datasets

Open datasets are those under a free and permissive license, and are able to be downloaded with requesting permission or signing an academic license. The open datasets used in this study are: *Canadian-French emotional* dataset (Gournay et al., 2018), *Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D)* (Cao et al., 2014), *EMO-DB* (Burkhardt et al., 2005), *eNTERFACE* dataset (Martin et al., 2006), *JL corpus* (James et al., 2018), *Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)* (Livingstone and Russo, 2018), *Sharif Emotional Speech Database (ShEMO)* (Mohamad Nezami et al., 2019), and the *Toronto Emotional Speech Set (TESS)* (Dupuis and Pichora-Fuller, 2011).

3.2 Licensed Datasets

Licensed datasets are those that require signing an academic or other license in order to gain access to the data. The licensed datasets used in this study are: *Database of Elicited Mood in Speech (DEMoS)* (Parada-Cabaleiro et al., 2019), *EmoFilm* (Parada-Cabaleiro et al., 2018), *Interactive Emotional Dyadic Motion Capture (IEMOCAP)* (Busso et al., 2008), *MSP-IMPROV* (Busso et al., 2017), *Surrey Audio-Visual Expressed Emotion (SAVEE)* database (Haq et al., 2008), and the *SmartKom* corpus, public set (Schiel et al., 2002).

4 Methodology

4.1 Models

We implement four neural network models that have been proposed in previous literature. These

Model	Input	CNN	RNN	Att.	# Params
Aldeneh	2D	✓			4.4M–13.7M
Latif	1D	✓	✓		1.2M
Zhang	1D	✓	✓	✓	0.7M
Zhao	2D	✓	✓	✓	0.6M

Table 2: Summary table of model parameters. CNN: convolutional neural network. RNN: recurrent neural network. Att.: attention pooling. The number of parameters for the Aldeneh model depends on the number of frequency bands in the input.

models were selected with the goal of having a variety of model types (convolutional and recurrent), variety of input formats (spectrogram and raw audio), and recency (within the past few years). After each model is introduced with citation, it will subsequently be referred to by the primary author's surname. A summary table of model types and number of parameters is given in Table 2. Each model outputs are probability distribution over N classes.

We implement the final model from Aldeneh and Mower Provost (2017). This consists of four independent 1D convolutions, followed by maxpooling. The resulting vectors are concatenated into a feature vector which is passed to two fully-connected layers. The Aldeneh model takes a 2D sequence of log Mel-frequency spectrograms as input.

The model from Latif et al. (2019) consists of 3 independent 1D convolutions of with batch normalisation and maxpooling. The filters are concatenated feature-wise and a 2D convolution is performed, again with batch normalisation and maxpooling. The final 1920-dimensional feature sequence is passed through a LSTM block, followed by 30% dropout and a fully-connected layer. The Latif model takes 1D raw audio as input.

The model from Zhao et al. (2019) consists of a convolutional branch and a recurrent branch that act on 2D spectrograms. The recurrent branch consists of a bidirectional LSTM with a single layer, whereas in the paper they used two layers. The convolutional branch consists of three sequential 2D convolutions, with batch normalisation, max-pooling and dropout. The filters and kernel sizes are different across convolutions and the resulting time-frequency axes are flattened and passed through a dense layer. The convolutional and recurrent branches are individually pooled using weighted attention pooling, concatenated and finally passed through a dense layer.

The model proposed in Zhang et al. (2019) acts on a raw audio waveform. The audio is framed with a frame size of 640 samples and shift of 160 samples. Two 1D convolutions with maxpooling are calculated along the time dimension. The features are then pooled in the feature dimension and flattened to a 1280-dimensional vector per frame. The sequences are fed into a 2-layer GRU, before weighted attention pooling, as in the Zhao model. Although this model was originally designed to perform multi-task discrete valence and arousal classification, we apply it to the single-task emotion label classification.

4.2 Cross-validation

We perform leave-one-speaker-out (LOSO) or leave-one-speaker-group-out (LOSGO) cross-validation for all tests. Before testing, we perform per-speaker standardisation of feature columns, as in (Schuller et al., 2009a). If a dataset has more than 12 speakers, then 6 random speaker groups are chosen for cross-validation. For IEMOCAP and MSP-IMPROV, each session defines a speaker group. All models are trained for 50 epochs with the Adam optimiser (Kingma and Ba, 2017) and a learning rate of 0.0001. The batch size used for the Aldeneh and Latif models was 32, for the Zhao model was 64, and for the Zhang model was 16. Each was trained using sample weights inversely proportional to the respective class sizes, so the each class had equal total weight. The sample weights were used to scale the cross-entropy loss. The metric reported is 'unweighted average recall' (UAR), which is simply the mean of the per-class recall scores. This incorporates all classes equally even if there is a large class bias, and minimises the effect of class distribution on the reported accuracy, so that models can't simply optimise for the majority class. Each test is repeated 3 times and averaged, except for the Zhang model, which was only tested once, because it took too long to train.

All models were implemented in Python using

the TensorFlow[2] Keras API. Testing was run on a machine with 64GB of RAM, an AMD Ryzen 3900X CPU, and two NVIDIA GeForce RTX 2080 Super GPUs, each with 8GB of VRAM. Each training run used only one GPU, however.

For the Zhang et al. (2019) and Latif et al. (2019) models we use the raw time domain signals. These are clipped to a maximum length of 80,000 samples (5 seconds at 16,000 kHz), but not padded, unlike the fixed spectrograms. For the Zhao et al. (2019) model we input a 5 second log-mel spectrogram with 40 mel bands calculated using a frame size of 25ms and frame shift of 10ms. Audio is clipped/padded to exactly 5 seconds. For the Aldeneh and Mower Provost (2017) model we test three different inputs: a 5 second 240 mel band spectrogram, 240 log-mel bands without clipping/padding, and 40 log-mel bands without clipping/padding. The log-mel bands are variable length sequences and are length-padded to the nearest larger multiple of 64, before batching. This way the models train with different sequence lengths.

5 Results

A table of results is given in Table 3 below. All combinations of dataset and model+features were tested. For comparison, we also report on the performance of the 'IS09' standard feature set introduced in the first INTERSPEECH emotion competition (Schuller et al., 2009b). For this we use a support vector machine (SVM) with radial basis function (RBF) kernel, with SVM parameter C and kernel parameter γ optimised using LOS(G)O cross-validation. We also report human accuracy where it has either been mentioned in the corresponding citation, or can be calculated from multiple label annotations provided with the dataset.

6 Discussion

From the results we see that the models using raw audio as input tend to perform worse than those using spectrogram input. There are also cases, such as on the Portuguese dataset, where the Zhang model performs the best of the four, and such as on the JL corpus, where the raw audio models are better than the fixed-size spectrogram models but worse than the variable length log-mel models.

There are many possible reasons for this, and due to time constraints, more thorough investigation was not able to be done. One reason is likely the lack of hyperparameter tuning. Hyperparameters like number of training epochs, learning rate, batch size, and model specific hyperparameters such as the number of convolution kernels or number of LSTM units, can have a moderate effect on the performance of each model. These would need to be optimised per-dataset using cross-validation, before testing. Another possible reason is the tendency for models to overfit. We found that the raw audio models were overfitting quite badly and achieving worse performance on the test set as a result, even though they have a moderate number of parameters. Regularisation techniques can help with this, such as dropout and regularisation loss, along with batch normalisation. Finally, while we tried to make our models as similar as possible to the original papers, there are likely implementation differences that negatively influence the performance of our models. The design of the Zhang model was for discrete arousal/valence prediction, and it is likely that a slightly modified architecture would better suit categorical emotion prediction. The other models were also tested with slightly different methodologies from ours, which would influence difference in reported results.

We also see a dependence on both dataset and features used. The Aldeneh model with 240 log-mels tended to be better than with only 40 log-mels, but also better than a fixed size 240 mel-band spectrogram, but this was dependent on dataset. It's possible that the zero-padding and -60dB clipping of the spectrograms negatively impacted the performance. The Zhao model performs best out of the four on the SmartKom dataset, achieving a UAR better than chance level, but still worse than the SVM with IS09 features. It's possible that in this instance the separate LSTM and convolutional branches have a greater effect. Unfortunately we were not able to test all combinations of spectrogram features with the Zhao model. In future we aim to complete this, as well as compare using spectrograms with different frame size and clipping parameters.

Finally, the time taken to train these models is quite long due to using full cross-validation. An argument can be made for predefined training/validation/test sets of larger datasets, but these are often created ad hoc and can vary between studies, so collective agreement would be needed for using these as a common standard.

[2] https://www.tensorflow.org/

Corpus	A1	A2	A3	L	N	O	SVM-IS09	Human
CaFE	53.8	54.0	52.1	22.3	32.3	48.0	**57.2**	
CREMA-D	66.6	**67.0**	63.4	42.4	48.4	57.9	65.0	40.0
DEMoS	61.4	**61.9**	61.5	25.5	26.9	45.7	51.2	61.1
EMO-DB	73.2	74.6	72.7	45.2	49.7	53.7	**82.1**	84.3
EmoFilm	49.6	49.7	49.4	40.2	45.6	44.7	**53.2**	73
eNTERFACE	77.9	**79.4**	77.4	38.6	45.0	66.4	76.3	
IEMOCAP	**61.1**	60.5	58.2	46.2	49.2	58.3	59.8	73.8
JL	65.8	**67.8**	47.9	54.0	61.2	46.6	66.2	69.1
MSP-IMPROV	47.2	47.5	46.2	35.2	38.0	48.6	**52.4**	77.8
Portuguese	38.3	39.0	41.5	37.4	43.3	39.9	**50.0**	73.2
RAVDESS	32.5	39.5	60.0	29.6	32.9	43.0	**60.6**	62.5
SAVEE	58.4	**59.6**	48.5	34.8	33.0	30.1	57.0	66.5
ShEMO	54.6	**55.7**	50.7	43.6	48.4	51.8	51.3	
SmartKom	15.8	16.8	17.5	16.0	16.7	22.6	**28.5**	
TESS	48.7	49.5	**55.1**	38.5	30.6	48.4	45.9	82

Table 3: Table of results. All values are given in UAR. A1: Aldeneh model with variable 40 log-mels. A2: Aldeneh model with variable 240 log-mels. A3: Aldeneh model with fixed 5s 240-mel spectrogram. L: Latif model with 5s raw audio. N: Zhang model with 5s raw audio. O: Zhao model with fixed 5s 40-mel spectrogram. Human accuracy is the average accuracy of a human rater, either tested in the relevant citation, or calculated directly from annotations (e.g. CREMA-D).

7 Conclusion

In this paper we have presented an evaluation of different neural network models proposed for emotion recognition, and compared their performance for discrete emotion classification on 15 publicly available and academic datasets. We used a consistent methodology across all datasets, and have kept hyperparameters very similar across the proposed models. The results show differences in the performance of the models which sometimes depends on the evaluated dataset. We also showed that the models requiring raw audio input tended to perform worse than the ones requiring spectrogram input, however more testing is required, with hyperparameter tuning and regularisation techniques, to determine the cause of this performance difference. In general, our work serves as a baseline for comparison for future research.

In future, we aim to additionally test models using utterance level features as input, and compare with non-neural network models such as SVM and random forests. We also aim to test feature generation methods such as bag-of-audio-words and unsupervised representation learning.

Acknowledgements

The authors would like to thank the University of Auckland for funding this research through a PhD scholarship. We would like to thank in particular the School of Computer Science for providing the computer hardware to train and test these models. We would also like to thank the anonymous reviewers who submitted helpful feedback on this paper.

References

Zakaria Aldeneh and Emily Mower Provost. 2017. Using regional saliency for speech emotion recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2741–2745. ISSN: 2379-190X.

Felix Burkhardt, Astrid Paeschke, Miriam Rolfes, Walter F. Sendlmeier, and Benjamin Weiss. 2005. A database of German emotional speech. In *Interspeech 2005 - Eurospeech*, pages 1517–1520, Lisbon, Portugal.

C. Busso, S. Parthasarathy, A. Burmania, M. AbdelWahab, N. Sadoughi, and E. Mower Provost. 2017. MSP-IMPROV: An Acted Corpus of Dyadic Interactions to Study Emotion Perception. *IEEE Transactions on Affective Computing*, 8(1):67–80.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. 2008. IEMOCAP: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4):335.

H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma. 2014. CREMA-

D: Crowd-Sourced Emotional Multimodal Actors Dataset. *IEEE Transactions on Affective Computing*, 5(4):377–390.

Kate Dupuis and M. Kathleen Pichora-Fuller. 2011. Recognition of emotional speech for younger and older talkers: Behavioural findings from the toronto emotional speech set. *Canadian Acoustics*, 39(3):182–183.

Philippe Gournay, Olivier Lahaie, and Roch Lefebvre. 2018. A canadian french emotional speech dataset. In *Proceedings of the 9th ACM Multimedia Systems Conference*, pages 399–402. ACM.

Sanaul Haq, Philip JB Jackson, and James Edge. 2008. Audio-visual feature selection and reduction for emotion classification. In *International Conference on Auditory-Visual Speech Processing 2008*, pages 185–190, Tangalooma Wild Dolphin Resort, Moreton Island, Queensland, Australia.

Jesin James, Li Tian, and Catherine Inez Watson. 2018. An Open Source Emotional Speech Corpus for Human Robot Interaction Applications. In *Interspeech*, pages 2768–2772.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*. ArXiv: 1412.6980.

Shashidhar G. Koolagudi and K. Sreenivasa Rao. 2012. Emotion recognition from speech: a review. *International Journal of Speech Technology*, 15(2):99–117.

Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, and Julien Epps. 2019. Direct Modelling of Speech Emotion from Raw Speech. *arXiv:1904.03833 [cs, eess]*. ArXiv: 1904.03833.

Steven R. Livingstone and Frank A. Russo. 2018. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLOS ONE*, 13(5):e0196391.

O. Martin, I. Kotsia, B. Macq, and I. Pitas. 2006. The eNTERFACE' 05 Audio-Visual Emotion Database. In *22nd International Conference on Data Engineering Workshops*, pages 8–8.

Omid Mohamad Nezami, Paria Jamshid Lou, and Mansoureh Karami. 2019. ShEMO: a large-scale validated database for Persian speech emotion detection. *Language Resources and Evaluation*, 53(1):1–16.

Emilia Parada-Cabaleiro, Giovanni Costantini, Anton Batliner, Alice Baird, and Björn Schuller. 2018. Categorical vs Dimensional Perception of Italian Emotional Speech. In *Interspeech 2018*, pages 3638–3642. ISCA.

Emilia Parada-Cabaleiro, Giovanni Costantini, Anton Batliner, Maximilian Schmitt, and Björn W. Schuller. 2019. DEMoS: an Italian emotional speech corpus. *Language Resources and Evaluation*.

Christian Peter and Russell Beale, editors. 2008. *Affect and Emotion in Human-Computer Interaction: From Theory to Applications*. Number 4868 in Lecture Notes in Computer Science. Springer Science & Business Media. Google-Books-ID: BwdcWtO666EC.

Florian Schiel, Silke Steininger, and Ulrich Türk. 2002. The SmartKom Multimodal Corpus at BAS. In *Third International Conference on Language Resources and Evaluation*, pages 200–206, Las Palmas, Canary Islands, Spain. Citeseer.

B. Schuller, B. Vlasenko, F. Eyben, G. Rigoll, and A. Wendemuth. 2009a. Acoustic emotion recognition: A benchmark comparison of performances. In *2009 IEEE Workshop on Automatic Speech Recognition Understanding*, pages 552–557.

Björn Schuller, Stefan Steidl, and Anton Batliner. 2009b. The INTERSPEECH 2009 emotion challenge. In *10th Annual Conference of the International Speech Communication Association*, pages 312–315, Brighton, United Kingdom.

A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, G. Meier, and B. Schuller. 2011. Deep neural networks for acoustic emotion recognition: Raising the benchmarks. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5688–5691.

Z. Zhang, B. Wu, and B. Schuller. 2019. Attention-augmented End-to-end Multi-task Learning for Emotion Prediction from Speech. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6705–6709.

Z. Zhao, Z. Bao, Y. Zhao, Z. Zhang, N. Cummins, Z. Ren, and B. Schuller. 2019. Exploring Deep Spectrum Representations via Attention-Based Recurrent and Convolutional Neural Networks for Speech Emotion Recognition. *IEEE Access*, 7:97515–97525.

Popularity Prediction of Online Petitions using a Multimodal Deep Regression Model

Kotaro Kitayama[♠] **Shivashankar Subramanian**[♡] **Timothy Baldwin**[♡]
♠ Tohoku University
♡ The University of Melbourne
kitayama@ecei.tohoku.ac.jp {shivashankar.subramanian,tbaldwin}@unimelb.edu.au

Abstract

Online petitions offer a mechanism for people to initiate a request for change and gather support from others to demonstrate support for the cause. In this work, we model the task of petition popularity using both text and image representations across four different languages. We evaluate our proposed approach using a dataset of 75k petitions from Avaaz.org, and find strong complementarity between text and images.

1 Introduction

A petition is a formal request for change or an action to any authority, co-signed by a group of supporters (Ergazakis et al., 2012). The targets of petitions are usually government agencies and business organizations.

In this work we study petitions from Avaaz.org, a popular petition platform available across six continents, with support for seventeen different languages. Avaaz provides a platform for petitions and funding campaigns. An example petition is given in Figure 1, opposing the Anti-Counterfeiting Trade Agreement (ACTA) and supporting a free and open internet. The popularity of a petition, in terms of the number of signatures it attracts, is critical to its success, and predicting popularity can help petition organizers to enhance their engagement strategy by optimising the petition content. In particular in this work, we target the task of predicting petition popularity at the time of submission (independent of any social media or direct signature signal).

While existing work on petitions has focused on their text content (Elnoshokaty et al., 2016; Subramanian et al., 2018), images are also a key ingredient.[1] Additionally, despite petitions being popular in many different languages, there has been no work on multilingual modeling. From a social science viewpoint, multilingual analysis can contribute to an understanding of issues present in different languages (or regions).

Previous research has shown that, other than petition content, metadata is also effective in modelling its popularity. Elnoshokaty et al. (2016) showed that the category of a petition has an influence on its popularity and success, e.g., *human trafficking* related petitions get more signatures than *health* related ones. Vidgen and Yasseri (2019) investigated the interaction between topics and geographic features, and showed that some issues receive broad geographic support (e.g., *law & order*, *work & play*) but others are far more local (e.g., *animals & nature*, *driving*). Since we aim to model petition popularity across multiple countries, we study the utility of the author's country information.

Our contributions in this work are as follows: (1) we propose a multimodal regression approach for petition popularity prediction task using text and image features; (2) we experiment in both mono- and multi-lingual data settings, to evaluate the impact of training data from other languages; and (3) we develop a novel multimodal, multilingual dataset for the task.[2]

2 Related Work

The majority of work on modeling petition popularity has focused on predicting popularity growth over time based on an initial popularity trajectory (Hale et al., 2013; Yasseri et al., 2017; Proskurnia et al., 2017), e.g. given the number of signatures a petition gets in the first x hours, predicting the total number of signatures at the end of its lifetime. Since the popularity of a petition also depends on its author's campaign strategies, Asher et al. (2017)

[1] https://secure.avaaz.org/en/community_petitions/how_to_create_petition/

[2] The dataset and all code associated with this paper will be available at: https://github.com/kkitayama/petitions-with-image.

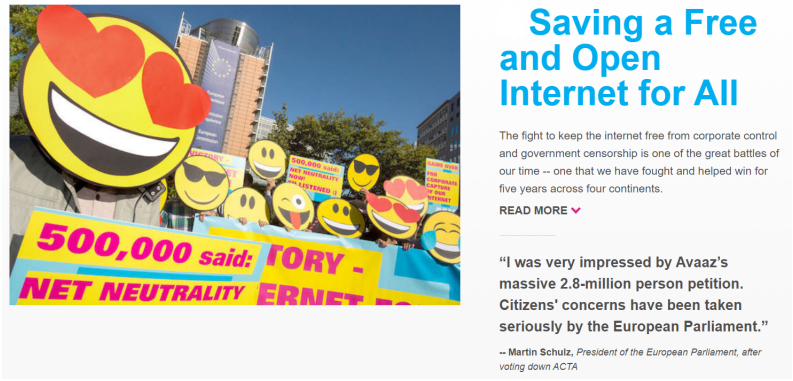


Figure 1: An example petition from Avaaz, which led to the European Parliament abandoning ACTA.

and Proskurnia et al. (2017) examined the impact of sharing petitions on Twitter, as a time series regression task. However, none of this work analyzed the petition's content, which is a primary focus in this work, in addition to making the prediction at the time of *submission* rather than based on early social indicators.

Elnoshokaty et al. (2016) analyzed Change.org petitions by performing correlation analysis of popularity against the petition's category, target goal set, and the distribution of words in General Inquirer categories (Stone et al., 1962). Subramanian et al. (2018) is the closest work to this paper, which targets UK and US government petitions, and poses popularity prediction as a text regression task using a convolutional neural network model.

Though petition platforms such as Avaaz.org and Change.org are popular in different countries and languages, almost all the existing work has focused on the monolingual setting (almost exclusively on English). With the increasing use of petitions across the globe, it is necessary to model petitions across different languages (Aragón et al., 2018). Lastly, in addition to textual content, the choice of images and other multimodal information has been shown to have impact on the popularity of social media posts (Meghawat et al., 2018; Wang et al., 2018; Chen et al., 2019), but not utilized in the context of online petitions.

3 Dataset

We use petitions from the Avaaz.org dataset of Aragón et al. (2018), which consists of over 360k petitions in more than fifty languages. Each petition is made up of its textual content, author details, country, count of shares on social media, language, and other metadata. For our work, we use the top four languages based on the raw count of petitions: English, French, Portuguese, and Spanish. We extended the filtered dataset by crawling the image content for the petitions in those four languages from Avaaz.org. We removed petitions with empty content or where there is not a single majority language for all sentences (based on langid.py: Lui and Baldwin (2012)). The resulting dataset has a total of around 75k petitions, nearly 45% of the which have default images.[3] The distribution across languages is given in Table 1.

	Train	Dev	Test
English	14,262	1,800	1,800
Portuguese	21,888	2,700	2,700
French	13,647	1,700	1,700
Spanish	10,118	1,300	1,300

Table 1: Data split across languages

4 Methodology

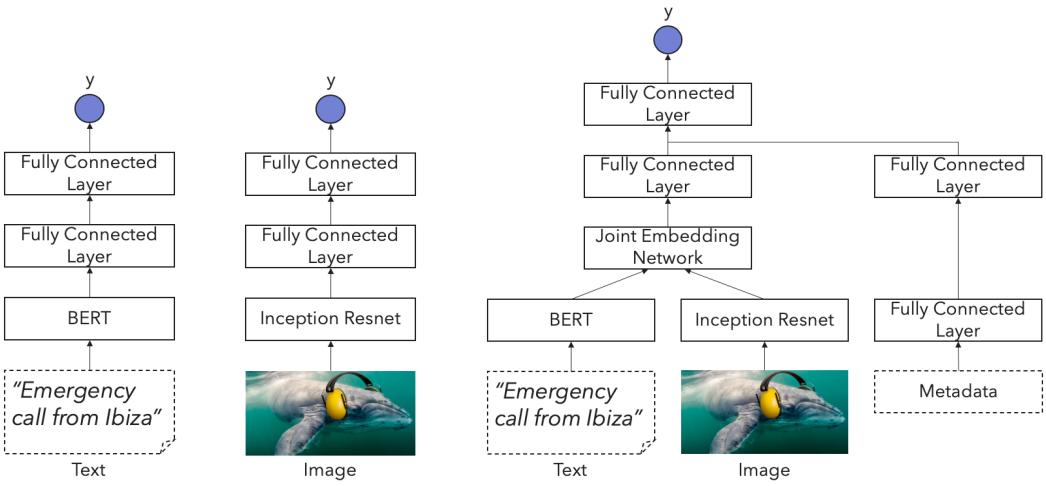
4.1 Model

We evaluate three classes of model: text-only, image-only, and combined text, image and metadata. In each case, we regress over the petition signatures, and use fully-connected hidden layers with a ReLU activation function before the final output layer. Note that we log-transform the signature count, consistent with previous work (Elnoshokaty et al., 2016; Proskurnia et al., 2017; Subramanian et al., 2018).

4.1.1 Text-only model

We employ two different text-only model architectures: (1) a CNN regression model (Bitvai and

[3] https://avaazdo.s3.amazonaws.com/do_generic*



(a) Text-only (b) Image-only (c) Combined model

Figure 2: Overview of the models, where y denotes the signature count.

Cohn, 2015) based on the method of Subramanian et al. (2018); and (2) a BERT (Devlin et al., 2019) regression model, where the `[CLS]` encoding of the final layer is used as the text representation.

For our monolingual experiments over English petitions, for the CNN model we use GloVe (Pennington et al., 2014) word embeddings, and for BERT we use the pre-trained BERT-base English model (Devlin et al., 2019). For the multilingual experiments, we use the pre-trained multilingual BERT model ("mBERT": Devlin et al. (2019)). An overview of the text model architecture is presented in Figure 2a.

4.1.2 Image-only model

For the image-only model, we use Inception-ResNet v2 (Szegedy et al., 2017) pre-trained on ImageNet, and extract the image representation from the penultimate layer. An overview of the image-only model is presented in Figure 2b.

4.1.3 Combined model

In the combined model, we use text, image, and metadata features, as detailed in Figure 2c, adopting the approach of Wang et al. (2018). We use text features extracted from the text-only model based on CNN or BERT, and image features from the image-only model. In both cases, we freeze all model parameters. Following that, text and image features are jointly embedded using a fully connected layer, a ReLU activation layer, a second fully connected layer, a batch normalization layer, and an L2 normalization layer (referred to as Joint Embedding Network in Figure 2c). Finally, the joint embedding and metadata embeddings are combined together using a fully connected layer.

As metadata, we use author's country information from the original dataset, encoded as a one-hot vector. Although the original data includes other metadata such as social media likes, we do not use it as it would not be available at the time of authoring.

4.2 Loss

We evaluate two types of regression loss functions. First we employ mean squared error in log-space ("`MSLE`") as used by Subramanian et al. (2018), and calculated as:

$$\frac{1}{N}\sum_{i=1}^{N}(\log(y_i+1)-\log(\hat{y}_i+1))^2$$

where y_i is the actual signature count and $\hat{y}_i$ is the predicted signature count. Second, we use mean absolute percentage error, again in log-scale ("`MAPE`"), jointly with `MSLE`. `MAPE` helps to capture the deviation between actual and predicted values, relative to the actual ones. An intuitive reason to use `MAPE` is to directly capture the expected model behavior consistent with the evaluation metric (see Section 5). The joint loss is computed as follows:

$$\frac{1}{N}\sum_{i=1}^{N}(\log(y_i+1)-\log(\hat{y}_i+1))^2 + k\times 100\times\frac{1}{N}\sum_{i=1}^{N}\left|\frac{\log(y_i+1)-\log(\hat{y}_i+1)}{\log(y_i+1)}\right|$$

where y_i is the actual signature count and $\hat{y}_i$ is the predicted signature count; k is a hyper-parameter,

	MSLE		MSLE + MAPE	
	MAPE↓	ρ↑	MAPE↓	ρ↑
IMG	50.6	0.235	37.1	0.200
TXT_{CNN}	40.9	0.363	36.0	0.354
TXT_{CNN} +IMG	39.7	0.392	35.6	0.375
TXT_{CNN} +IMG +CN	**39.2**	0.381	35.9	0.360
TXT_{BERT}	42.4	0.385	35.1	0.375
TXT_{BERT} +IMG	40.7	**0.405**	35.4	0.388
TXT_{BERT} +IMG +CN	41.1	0.397	**34.9**	**0.403**

Table 2: Monolingual English results; best results in bold. "TXT" = text, "IMG" = image, "CN" = country.

	MSLE		MSLE + MAPE	
	MAPE↓	ρ↑	MAPE↓	ρ↑
IMG	49.2	0.234	36.7	0.212
$\text{TXT}_{\text{mBERT}}$	46.8	0.277	**35.8**	0.239
$\text{TXT}_{\text{mBERT}}$ +IMG	**43.6**	**0.306**	36.3	0.298
$\text{TXT}_{\text{mBERT}}$ +IMG +CN	44.0	0.305	36.8	**0.300**

Table 3: Multilingual results (all languages); best results in bold. "Txt" = text, "IMG" = image, "CN" = country.

tuned on the development data.

5 Experiments

5.1 Settings

We use the predefined training/validation/test random splits (Table 1) for the experiments. We use mean absolute percentage error (MAPE) and Spearman's rank correlation as evaluation metrics, as commonly used by other popularity prediction tasks (Subramanian et al., 2018; Wang et al., 2018). For MAPE, lower is better, and a perfect system will score 0%; and for Spearman's rank correlation ("ρ") higher is better, and a perfect system will score 1.0.

For the CNN model, we set the number of filters to 100 and the dimensionality of the fully connected layer to 100, and for BERT monolingual and multilingual models, we set the dimensionality of the hidden layers to 300 and 500, respectively. For the image-only model (IMG), we set the dimensionality of the fully connected layer to 500 and 200 in the monolingual and multilingual models, respectively. All hyper-parameters were tuned on the development data.

5.2 Results

First, we evaluate the models using English data, and present the results in Table 2. For text-based modelling ("TXT"), we use CNN and BERT. For image-based modeling ("IMG"), we use ResNet encodings. From the results, it is evident that TXT is more discriminative than IMG, which provides the best standalone performance. But the combined text and image model performs better than the text-only model. In terms of the two loss functions, Spearman's ρ is largely the same as with simple MSLE, but the combined MSLE + MAPE loss predictably leads to substantial improvements in MAPE, especially for the image-only model. The inclusion of country data ("+CN") leads to marginal improvements.

	MSLE		MSLE + MAPE	
	MAPE↓	ρ↑	MAPE↓	ρ↑
English	48.7	0.246	35.6	0.276
Portuguese	47.2	0.280	35.6	0.276
French	43.4	0.331	35.9	0.258
Spanish	47.6	0.199	35.2	0.170

Table 4: Multilingual results (per language), based on $\text{TXT}_{\text{mBERT}}$ +IMG

In the multilingual setting, we use multilingual BERT ("mBERT") to represent text for all the languages. mBERT trains a single BERT model for over 100 languages with a large shared vocabulary (Devlin et al., 2019). Employing monolingual BERT trained on each language, as well as using cross-lingual language models (Conneau and Lample, 2019) are valid alternate approaches, which are left for future work. Here, we observe a similar overall trend where text-only models are slightly better than image-only models, and despite the almost fourfold increase in training data, the absolute results are worse than the monolingual results (Table 2) in the case of English in Table 4. In terms of loss, the performance for French with MSLE is superior to other languages. Lastly, Spearman's ρ for Spanish is quite a bit lower than the other languages, a result which requires further analysis.

6 Conclusions

We proposed an multimodal, multilingual approach to the task of petition popularity prediction, and found that while the text-only model was superior to the image-only model, the combination of multimodal features performed the best. Exploring further choices of metadata, and alternate ways to model multilingual text is left to future work.

References

Pablo Aragón, Diego Sáez-Trumper, Miriam Redi, Scott A. Hale, Vicenç Gómez, and Andreas Kaltenbrunner. 2018. Online petitioning through data exploration and what we found there: A dataset of petitions from avaaz.org. In *ICWSM-18 - 12th International AAAI Conference on Web and Social Media*.

Molly Asher, Cristina Leston Bandeira, and Viktoria Spaiser. 2017. Assessing the effectiveness of e-petitioning through Twitter conversations. *Political Studies Association (UK) Annual Conference*.

Zsolt Bitvai and Trevor Cohn. 2015. Non-linear text regression with a deep convolutional neural network. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 180–185.

Junhong Chen, Dayong Liang, Zhanmo Zhu, Xiaojing Zhou, Zihan Ye, and Xiuyun Mo. 2019. Social media popularity prediction based on visual-textual features with xgboost. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2692–2696.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ahmed Said Elnoshokaty, Shuyuan Deng, and Dong-Heon Kwak. 2016. Success factors of online petitions: Evidence from change.org. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 1979–1985.

Kostas Ergazakis, Dimitrios Askounis, Panagiotis Kokkinakos, and Anastasios Tsitsanis. 2012. An integrated methodology for the evaluation of epetitions. In *Empowering Open and Collaborative Governance*, pages 39–59. Springer.

Scott A Hale, Helen Margetts, and Taha Yasseri. 2013. Petition growth and success rates on the UK No. 10 Downing Street website. In *Proceedings of the 5th Annual ACM Web Science Conference*, pages 132–138.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

Mayank Meghawat, Satyendra Yadav, Debanjan Mahata, Yifang Yin, Rajiv Ratn Shah, and Roger Zimmermann. 2018. A multimodal approach to predict social media popularity. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 190–195.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Julia Proskurnia, Przemyslaw A. Grabowicz, Ryota Kobayashi, Carlos Castillo, Philippe Cudré-Mauroux, and Karl Aberer. 2017. Predicting the success of online petitions leveraging multidimensional time-series. In *Proceedings of the 26th International Conference on World Wide Web*, pages 755–764.

Philip J. Stone, Robert F. Bales, J. Zvi Namenwirth, and Daniel M. Ogilvie. 1962. The General Inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information. *Systems Research and Behavioral Science*, 7(4):484–498.

Shivashankar Subramanian, Timothy Baldwin, and Trevor Cohn. 2018. Content-based popularity prediction of online petitions using a deep regression model. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 182–188.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4278–4284.

Bertie Vidgen and Taha Yasseri. 2019. What, when and where of petitions submitted to the UK government during a time of chaos. *arXiv preprint arXiv:1907.01536*.

Ke Wang, Mohit Bansal, and Jan-Michael Frahm. 2018. Retweet wars: Tweet popularity prediction via dynamic multimodal regression. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1842–1851.

Taha Yasseri, Scott A Hale, and Helen Z Margetts. 2017. Rapid rise and decay in petition signing. *EPJ Data Science*, 6(1):1–13.

Exploring Looping Effects in RNN-based Architectures

Andrei Shcherbakov[μ] **Saliha Muradoğlu**[ΩΦ] **Ekaterina Vylomova**[μ]
[Ω]The Australian National University (ANU) [μ]The University of Melbourne
[Φ]ARC Centre of Excellence for the Dynamics of Language (CoEDL)
sandreas@unimelb.edu.au, saliha.muradgolu@anu.edu.au
ekaterina.vylomova@unimelb.edu.au

Abstract

The paper investigates *repetitive loops*, a common problem in contemporary text generation (such as machine translation, language modelling, morphological inflection) systems. We hypothesized that a model's failure to distinguish respective latent states for different positions in an output sequence may be the primary cause of the looping. Therefore, we propose adding a position-aware discriminating factor to the model in attempt to reduce that effect. We conduct a study on neural models with recurrent units by explicitly altering their decoder internal state. We use a task of morphological reinflection as a proxy to study the effects of the changes. Our results show that the probability of the occurrence of repetitive loops is significantly reduced by introduction of an extra neural decoder output. The output should be specifically trained to produce gradually increasing value upon generation of each character of a given sequence. We also explored variations of the technique and found that feeding the extra output back to the decoder amplifies the positive effects.

1 Introduction

Over the last few years we witnessed a significant progress in the field of Natural Language Processing (NLP). Many state-of-the-art models are based on neural architectures with recurrent units. For instance, Sutskever et al. (2014) proposed one of the first neural machine translation models that achieved results comparable with statistical models. Similarly, Plank et al. (2016) introduced a neural POS tagging model as a new state-of-the-art on the task. Recently, neural architectures almost superseded non-neural (finite-state or rule-based) approaches in morphology modelling tasks such as morphological reinflection (Cotterell et al., 2016, 2017) with average accuracy being over 90% on high-resource languages. Error analysis conducted by Gorman et al. (2019) demonstrated that among general misprediction errors such as syncretism, the models also produce certain "silly" errors that human learners do not make. One case of such errors, a looping error, is particularly notable. This type of error is not specific to the task and several other papers reported a similar problem (Holtzman et al., 2019; Vakilipourtakalou and Mou, 2020). Still, the causes and the nature of the error remains under-studied. Here we provide some insights on the causes of the issues and possible remedy to it. We consider morphological reinflection task for our experiments since it has low time and space requirements and, therefore, allows us to reproduce cases of looping in sufficient quantities and analyse them relatively easy.

2 Morphological reinflection task

Morphological inflection is the task of generating a target word form (e.g., "runs") from its lemma ("to run") and a set of target morphosyntactic features (tags, "Verb;Present Tense;Singular;3rd Person"). The task is called morphological *reinflection* when the lemma form is replaced with any other form and, optionally, its morphosyntactic features. This is a type of string-to-string transduction problem that in many cases pre-supposes nearly monotonic alignment between the strings. Traditionally, researchers either hand-engineered (Koskenniemi, 1983; Kaplan and Kay, 1994) or used trainable (Mohri, 1997; Eisner, 2002) finite state transducers to solve the task. Most recently, neural models were shown to outperform most non-neural systems, especially in the case of high-resource languages (Cotterell et al., 2016; Vylomova et al., 2020).

3 Data

In terms of the study we focus on two typologically diverse languages, Nen (Evans and Miller, 2016; Evans, 2017, 2019) and Russian. Nen is a Papuan language of the Morehead-Maro (or Yam) family, spoken in the Western province of Papua New Guinea by approximately 400 people. The language is highly under-resourced, and Muradoglu et al. (2020) is the only computational work on it we are aware of, and in current study we use the data derived from their corpus.

Russian, a Slavic language from Indo-European family, on the other hand, is considered as high-resource. We use the splits from the SIGMORPHON–CoNLL 2017 shared task on morphological reinflection (Cotterell et al., 2017).

We used medium sized training sets which occurred to yield highest rates of looped sequences in predicted word forms. The number of samples in the datasets are presented in Table 1.

	Nen	Russian
Training samples	1589	1000
Development samples	227	1000

Table 1: Dataset sizes

4 Experiments

We reused the hard attention model specifically designed for the morphological reinflection task (Aharoni and Goldberg, 2017) for our explorations. The model uses an external aligner (Sudoh et al., 2013) to extract input-to-output character sequence transformation steps for a given morphological sample. Instances of a special character (STEP) are inserted into transformed words to represent alignment step advances. The resulting seq2seq model is trained to perform transformation from a given lemma into a target inflected form which contains STEP characters. The model consists of two modules; (1) an array of LSTM (Hochreiter and Schmidhuber, 1997) encoders and (2) an LSTM decoder. When a STEP character occurs in a target sequence (either learnt or predicted), the encoder array index advances to the next position. It corresponds to advancing current pointer in the lemma by one character. In such a way, a hard monotonic attention schema is implemented.

In our experiments we computed counts of looped sequences in generated word forms during model evaluation rounds that were carried out upon each epoch of model training. We distinguished a generated character sequence as looped if it satisfies both of the following conditions: (1) the sequence contains at least 3 repeated instances of some character subsequence at its very end, and (2) the total length of those repeated subsequences reaches at least 8 characters. While applying such a criterion, we considered predicted sequences in their alphabetical form, with all STEP characters stripped off.[1]

We hypothesized that the looping is primarily caused by merging of decoder states relevant to different word positions. Therefore, introduction of variables that are guaranteed to be different at distinct stages of output word form production should reduce looped prediction rate. Presence of such a variable would facilitate distinguishing states that correspond to different parts of generated word, if even closely surrounding character sequences are similar. To implement this idea, we introduced an extra decoder output that is trained to always be increasing while new output characters are produced. More specifically, we added an extra output r and an extra input $\tilde{r}$ to the decoder. To ensure that r increases gradually while target word characters are generated, we modified calculation of total loss in the model training, allowing an extra (hinge-like) term as follows:

$$L = \max(0, \gamma \cdot (s - \Delta r)) \quad (1)$$

Here Δr is the difference between current and previous r values. Initially, for every predicted word form r is set to zero. Having observed the dynamics of r value in preliminary training experiments, we chose $\gamma = 50; s = 0.05$.

For better exploration of different factors, we tested combinations of the following setting variations:

- Feeding r back to $\tilde{r}$ vs. leaving it unused (letting $\tilde{r} = 0$). We hypothesized that even when an increasing output itself isn't used, computation of its value still affects neural weights at the front layer of the decoder.
- Requiring r to increase vs. leaving it free.
- Scalar vs. vector r (in the latter case, terms according to equation 1 are to be added per each component).

[1] We didn't consider possible irregular (chaotic) looping cases as they are extremely rare.

- Using an externally provided auto-incremented value for r instead of an extra decoder output.

Table 2 presents mode denotations we use in the paper.

We repeated experiments 15 times for each distinct setting. The result figures presented are normalized to single experiment.

denotation	goal for r	$\tilde{r}$ value
n (*"**n**one"*)	none	zero
i (*"**i**ncrement"*)	r is ablated	incrementing
f (*"**f**eedback"*)	none	previous r
u (*"**u**nused"*)	increase	zero
s (*"all **s**et"*)	increase	previous r

Note: if r is a vector, its size is added before a mode symbol: '3f', '3u', '3s'.

Table 2: A summary of explored modes

mode	**nen**	**ru**	mode	**nen**	**ru**
n	0.040	2.313	i	0.020	0.033
	0	1.267		0	0
s	0.017	0.017	3s	0.030	**0.003**
	0	0		0.066	0
u	**0.010**	0.027	3u	0.810	39.87
	0	0.133		0	24.13
f	0.087	5.770	3f	5.823	107.2
	0.066	2.800		2.667	114.7

Table 3: Average looping counts (per epoch) observed at epochs 15..34

mode	**nen**	**ru**	mode	**nen**	**ru**
n	0.725	0.717	i	0.726	0.724
s	**0.732**	0.750	3s	0.716	**0.753**
u	0.727	0.728	3u	0.704	0.669
f	0.432	0.451	3f	0.668	0.574

Table 4: Development set accuracy achieved at different modes

5 Results

The plots given in Fig. 1 present counts of looped predictions at different epochs for the two datasets used (Nen and Russian).[2] It can be observed that

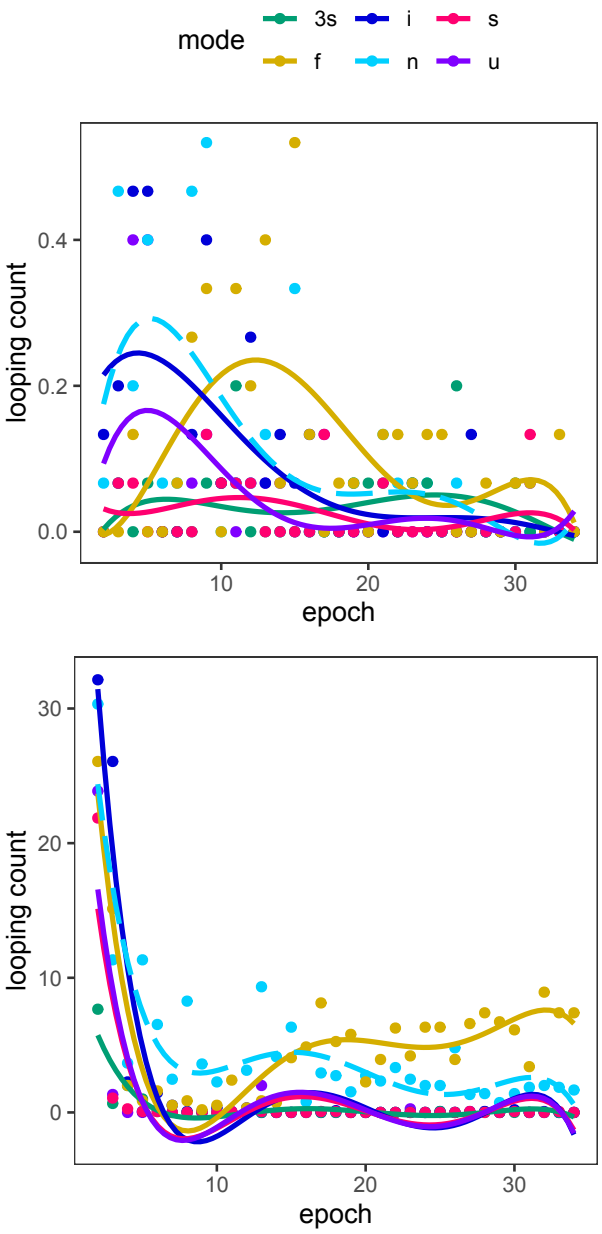


Figure 1: Looping counts observed in training a hard attention model on morphological datasets for Nen (upper plot) and Russian (lower plot)

training a model with increasing r (modes 's', '3s') demonstrates significantly lower rates of looped word generation compared to the baseline mode ('n'). This is true for almost all considered epochs. One may also note that the 'u' mode yields results comparable to ones obtained with the 's' mode. This fact means that the presence of gradually incrementing decoder output is helpful for fighting looping even when the output isn't used. However, if the output is free of constraints and is fed back to the decoder (mode 'f'), the effect is mostly

[2]The curves shown at Fig. 1, 2 are generated by a polynomial smoothing procedure from a dataset with high variance. They may expose some irrelevant artifacts, for example, they fall to negative count values at some points.

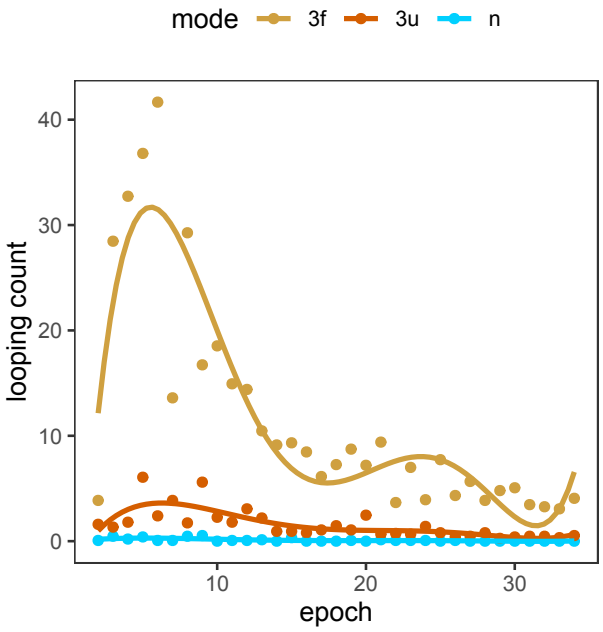


Figure 2: Looping count increase observed at some modes on a Nen language morphological dataset (see Figure 1 for other modes)

negative.

Fig. 1 demonstrates the results of the same kind for the modes that occur to be less looping-prone than the baseline mode. When its components weren't trained to gradually increase (mode '3f'), a vector of 3 feedback values drastically increased looping rate at all epochs. If a vector of 3 increasing components was produced but wasn't fed back as input, the results were still negative. This is surprising because the result for a respective scalar mode ('u') is positive.

Table 3 shows average looping counts for the 'later' epochs (15..34). Those epochs are more significant for the final quality assessment because maximum accuracy is usually achieved at one of them, so they have relatively high probability of producing the best model. Also, the table displays looping counts observed at epochs yielding best prediction accuracy as measured at a respective development set. The figures demonstrate that using modes with gradually increasing r ('s', '3s', 'u', 'i') yields significant reduction of looping rate. The only exception is mode '3u' which causes increase of the rate. As for the 'f' and, especially, '3f' modes (feeding an output back without requirement to grow), they may cause unacceptable high frequency of looped sequence generation. Overall, the digits are in line with the trends shown in the figures.

Increasing the dimensionality of extra decoder output sometimes yields an improvement ('3s' mode) but generally the results suggest that vector size is a factor causing looping rate increase. Finally, scalar seems to be more preferable than vector.

Table 4 shows prediction accuracy figures achieved in the experiments. For each training run, the epoch which produced the highest prediction accuracy against the development set was selected. Then, an average over repeated similar experiments was calculated. According to the figures, 's' mode yields a notable improvement of accuracy. In contrast, sticking to the 'f' mode causes a dramatic decline of accuracy.

6 Discussion

We have found a strong evidence that the presence of a decoder output which is trained to progressively increment reduces the average rate of looping sequences in multiple times. In most cases the positive effect is more significant if this output is fed back to the decoder, although there are exceptions of minor magnitude. Attempts to scale the effect further by increasing dimensionality of progressively incrementing variables are sometimes successful. Still, if we consider an average explored case, the mode 's' seems to be the most effective and consistent in fighting looping. We also observed that presence of an auto-incremented decoder input (mode 'i') leads to looping rate reduction, but the effect is superior if the decoder itself serves to produce a gradually increasing value. Thus, the practical recommendation arising from our research should be (1) adding an extra scalar output to the decoder, (2) endorsing it to increase by inclusion a respective term into a training loss formula, and (3) feeding it back as an encoder input.

Conceptually, it isn't surprising that the presence of an increasing variable helps the decoder to distinguish states rated to different phases of output word production and such a way reduces probability of falling into a loop. Still, the details of this mechanism yet need exploration. In our current work we made no attempt to enforce the usage of the new variable in any way; we only made such a usage potentially possible. A detailed exploration of its effect on the learning process is yet a subject of further research. And, what is even more

practically important, we yet need to find how the system design may be changed to incorporate progressive variables in a more explicit, controllable and efficient way.

The introduction of feedback variables adds elements of RNN architecture to the decoder. We observed highly negative results when such variable values weren't constrained (modes 'f' and, especially, '3f'). This indirectly suggests that RNN schema may not be a good solution for a decoder in terms of looping prevention.

7 Related Work

Holtzman et al. (2019) associated the problem with a more general *degeneration* issues that also includes production of blank and incoherent text. The authors observed that the issue appears during in maximization-based decoding methods such as beam search. As a remedy, they proposed a nucleus sampling technique that truncates unreliable tail of the probability distribution in the decoder part. Kulikov et al. (2019) also compared two search strategies, greedy and beam, proposing a novel iterative beam search strategy that increases diversity of the candidate responses. Contrary to that, Welleck et al. (2019) suggests that the problem cannot be solved by making beam search predictions more diverse. Instead, they propose focusing on likelihood loss, and introduce *"unlikelihood training"* that assigns lower probability to unlikely generations. Finally, following earlier observations on chaotic states w.r.t model parameters in Bertschinger and Natschläger (2004) and Laurent and von Brecht (2016), Vakilipourtakalou and Mou (2020) study chaotic behavior (Kathleen et al., 1996) in RNNs that are defined as iterative maps (Strogatz, 1994).

8 Conclusion

We proposed and explored a simple technique that reduces rate of repetitive loops occurrence in a neural decoder output. Our work was inspired by a hypothesis that looping effects in a neural decoder are caused by its inability to distinguish states related to different positions in a generated word. We both provided a simple and universal practical solution and outlined a promising direction for further research.

References

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015, Vancouver, Canada.

Nils Bertschinger and Thomas Natschläger. 2004. Real-time computation at the edge of chaos in recurrent neural networks. *Neural computation*, 16(7):1413–1436.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared Task—Morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22, Berlin, Germany. Association for Computational Linguistics.

Jason Eisner. 2002. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 1–8.

Nicholas Evans. 2017. Quantification in nen. In *Handbook of Quantifiers in Natural Language: Volume II*, pages 571–607. Springer.

Nicholas Evans. 2019. *Waiting for the Word: Distributed Deponency and the Semantic Interpretation of Number in the Nen Verb*, pages 100–123. Edinburgh University Press.

Nicholas Evans and Julia Colleen Miller. 2016. Nen. *Journal of the International Phonetic Association*, 46(3):331–349.

Kyle Gorman, Arya D. McCarthy, Ryan Cotterell, Ekaterina Vylomova, Miikka Silfverberg, and Magdalena Markowska. 2019. Weird inflects but OK: Making sense of morphological generation errors. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 140–151, Hong Kong, China. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

T Kathleen, D Tim, and A James. 1996. *CHAOS: an introduction to dynamical systems*. Springer, New York, NY, USA.

Kimmo Koskenniemi. 1983. *Two-level morphology: A general computational model for word-form recognition and production*, volume 11. University of Helsinki, Department of General Linguistics Helsinki, Finland.

Ilia Kulikov, Alexander Miller, Kyunghyun Cho, and Jason Weston. 2019. Importance of search and evaluation strategies in neural dialogue modeling. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 76–87.

Thomas Laurent and James von Brecht. 2016. A recurrent neural network without chaos. *arXiv preprint arXiv:1612.06212*.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational linguistics*, 23(2):269–311.

Saliha Muradoglu, Nicholas Evans, and Hanna Suominen. 2020. To compress or not to compress? a finite-state approach to Nen verbal morphology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 207–213, Online. Association for Computational Linguistics.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418.

Steven H Strogatz. 1994. Nonlinear dynamics and chaos: with applications to physics. *Biology, Chemistry and Engineering*, page 1.

Katsuhito Sudoh, Shinsuke Mori, and Masaaki Nagata. 2013. Noise-aware character alignment for bootstrapping statistical machine transliteration from bilingual corpora. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 204–209.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Pourya Vakilipourtakalou and Lili Mou. 2020. How chaotic are recurrent neural networks? *arXiv preprint arXiv:2004.13838*.

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrew Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, Christo Kirov, Garrett Nicolai, Adina Williams, Antonios Anastasopoulos, Hilaria Cruz, Eleanor Chodroff, Ryan Cotterell, Miikka Silfverberg, and Mans Hulden. 2020. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–39, Online. Association for Computational Linguistics.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Transformer Semantic Parsing

Gabriela Ferraro[1,2] and **Hanna Suominen**[2,1,3]

[1]Commonwealth Scientific and Industrial Research Organisation, Australia
[2]Research School of Computer Science, The Australian National University, Australia
[3]Department of Future Technologies, University of Turku, Finland
gabriela.ferraro@data61.csiro.au
hanna.suominen@anu.au

Abstract

In neural semantic parsing, sentences are mapped to meaning representations using encoder–decoder frameworks. In this paper, we propose to apply the Transformer architecture, instead of recurrent neural networks, to this task. Experiments in two data sets from different domains and with different levels of difficulty show that our model achieved better results than strong baselines in certain settings and competitive results across all our experiments.

1 Introduction

Semantic parsing maps natural language sentences to meaning representations including, but not limited to, logical formulas, Structured Query Language (SQL) queries, or executable codes. In recent years, end-to-end neural semantic parsing has achieved good results (Dong and Lapata, 2016; Jia and Liang, 2016a; Ling et al., 2016; Dong and Lapata, 2018; Finegan-Dollak et al., 2018). The main advantage of these models is that they do not require intermediate representations, lexicons, manually designed templates, or handcrafted features.

Current neural semantic parsing models use encoder–decoder architectures with Recurrent Neural Networks (RNNs). One drawback of RNNs is their inability to capture long distance relationships between input tokens or between input and output tokens (Yu et al., 2019). Hence, to model dependencies disregarding their distance, the best performing models also include some kind of an attention mechanism (Vaswani et al., 2017), which allows to model dependencies more accurately; specifically, the Transformer architecture introduced in this last reference has become the new state-of-the-art for sequence-to-sequence problems.

The Transformer architecture consists of a self-attention mechanism and does not include recurrent layers. Unlike RNNs, in which sequences are processed sequentially — word by word — Transformer models process the entire sentence as a whole. This characteristic is particularly beneficial in capturing long distance dependencies as the self-attention mechanism sees all the words at once. Despite the success of Transformer, to the best of our knowledge, prior to Li (2019) that founded our paper, this framework has never been applied to semantic parsing before. Thus, in this paper, we propose the Transformer architecture for semantic parsing.

A well-known limitation of sequence-to-sequence models is their inability to learn competitive parameter values for words that are rare in a given data set. To alleviate this problem, a common method is to anonymise entities with their respective types. For example, city names such as *Denver* are anonymised as *ci0* and later, as part of post-processing, put back in the output utterance. Neural semantic parsing models are usually trained and tested using data sets where variables are identified and anonymised beforehand — as in the example above — which considerably reduces the difficulty of the semantic parsing task (Finegan-Dollak et al., 2018). As a result, many input sentences of the test set are seen prematurely while training.

Consequently, we have a twofold approach to evaluate our model more extensively in this work and demonstrate its contributions to semantic parsing: First, we use non-anonymised versions of two data sets for semantic parsing from different domains, as well as different data splits. Second, we test the ability of our model to compose new output meaning representations. These experiments give evidence of the Transformer model outperforming strong baselines in certain settings. Its results are competitive on other settings across the data sets.

The rest of the short paper is organised as follows: Section 2 includes the related work in neu-

ral semantic parsing. Section 3 describes our model architecture, data sets, and experimental setup. Section 4 introduces our experimental results. Section 5 concludes our study.

2 Related Work

Encoder–decoder architectures based on neural networks have been applied in the past five years (i.e., up to, and including, 2019) to semantic parsing, but they have typically learnt from sentences paired with meaning representations without using explicit syntactic knowledge. Dong and Lapata (2016) have proposed two models with an attention mechanism as follows: the first model generates sequences and the second one generates trees as logical formulae that are hierarchically structured. Both models include an attention mechanism over a RNN that has the ability of focusing on a subset of input tokens or features. They also provide soft alignments between the input sentences and the logical formulae. Later, Dong and Lapata (2018) have proposed to use a two-step (coarse-to-fine) decoder to better model the compositionality of the logical formulae. Finally, to overcome the limitations of semantic parsing data sets being small and domain-dependent, several methods, such as multi-task learning (Susanto and Lu, 2017; Herzig and Berant, 2017; Fan et al., 2017), transfer learning (Kennardi et al., 2019; Damonte et al., 2019) , and data augmentation (Jia and Liang, 2016a; Kočiský et al., 2016), have been applied.

However, the aforementioned models cannot address the fact that the distance between tokens is not always an indication of a weak relationship. This problem becomes significantly worse for long sentences paired with long logical formulae. Thus, we propose to use a Transformer-based model in which token relations are not affected by the (long) distance.

As mentioned, data sets for semantic parsing are small and neural models do not tend to be good at learning the appropriate parameters for the long tail of rare words. To mitigate this problem, a common method is to apply variable or entity anonymisation as a pre-processing step. Later, the entities are put in the output sequence in a post-processing step. Another strategy is to use Pointer networks (Vinyals et al., 2015) where input tokens are copied to the output sequence at each decoding step. Moreover, attention-based copying (Jia and Liang, 2016b) refers to a mechanism in which the decoder can either choose to copy over a word to the output sequence or to pick from a softmax over the entire vocabulary. In our study, we use two data sets from different domains with and without variable anonymisation, and different splits to reflect diffing complexity levels of the semantic parsing task.

3 Methods

As our model architecture, we have implemented a self-attention neural semantic parsing model with the mechanism of Transformer (see Vaswani et al. (2017, Figure 1) for further information). As in other state-of-the-art sequence-to-sequence architectures, Transformer is essentially an encoder–decoder structure with blocks for encoding and decoding. Similar to other neural generation models with attention, the output of the last layer of the encoder is used as part of the input of each layer of the decoder. The most significant difference between Transformer and other sequence-to-sequence models is that Transformer uses neither *Convolutional Neural Networks* (CNNs) nor RNNs. Instead, it uses self-attention, which reduces path lengths within the network, thus minimising the loss of information due to computations.

Identical encoder blocks consist of multi-head self-attention and fully-connected feed-forward layers. After each layer there is a residual module, followed by a normalisation step. This produces a 512-dimensional output.

Each of identical decoder blocks has a multi-head self-attention layer, fully-connected feed-forward layer, and multi-head attention layer over the output of the encoder stack. Again, a residual module and normalisation step are applied to each output layer. Multi-head self-attention layers in encoding and decoding are similar, except in the latter one adds a mask operation between scaling and the softmax activation function. The reason for adding this look-ahead mask is to avoid that at the timestamp t, the tokens after t are used for predicting token at t .

In most Natural Language Processing tasks, the model should capture the order and position information from the sequential inputs. This is one of the advantages of RNNs and CNNs. Thus, our model includes a position embedding in the source and target inputs as in Vaswani et al.

Table 1: Example sentences and their corresponding logical formulae. Abbreviations: anon — anonymised, ground_transport — GT, quest — question,

Data set	Input	Output
ATIS	ground transport in ci0	(lambda $0 e (and GT) (to_city ci0)))
ATIS non-anon quest-split	ground transport in Denver	(lambda $0 e (and (GT $0) (to_city $0 denver)))
GEO	how many citizen in s0	(population:i s0)
GEO non-anon quest-split	how many citizens in Alabama	(population alabama)
GEO non-anon query-split	how many citizens in Boulder	(population boulder)

Table 2: Number of training (Train), development (Dev), and test (Test) examples

Data set	Train	Dev	Test
ATIS	4,434	491	448
ATIS non-anon	4,029	504	504
GEO	600	0	280
GEO non-anon quest-split	583	15	279
GEO non-anon query-split	543	148	186

(2017). Furthermore, word embeddings are randomly initialised for source and target inputs to treat them equally.

We use the Adam optimiser. The learning rate is set to 3×10^{-4}. The dimension of the self-attention model is $1,024$. From 6 to 8 encoder and decoder blocks are used. The dropout rate is 0.4 and maximum number of epochs 720 with early stopping.

We have used two semantic parsing data sets — namely ATIS with queries from a flight booking system (Price, 1990; Dahl et al., 1994; Zettlemoyer and Collins, 2007) and GEO with queries about US geographical information (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005) — for evaluation. The meaning representation of data sets is lambda calculus.

Table 3: Vocabulary (vocab) size for sentences and logical forms in the ATIS and GEO training sets. Entity anonymisation has a bigger impact in the vocab size of the input (I) than in the vocab size of the output (O).

Data set	I vocab	O vocab
ATIS	166	433
ATIS non-anon	444	887
GEO	51	120
GEO non-anon quest-split	141	243
GEO non-anon query-split	149	254

Table 4: The accuracy [%] on ATIS and GEO on anononymised test sets

Model	ATIS	GEO
Statistical Baselines		
ZC07 (Zettlemoyer and Collins, 2007)	84.6	86.07
TISP (Zhao and Huang, 2015)	84.2	**88.9**
Neural Baselines		
Seq2Seq + Attention (Dong and Lapata, 2016)	84.15	84.6
Seq2Tree + Attention (Dong and Lapata, 2016)	86.9	87.1
ASN (Rabinovich et al., 2017)	85.3	85.7
ASN + Attention (Rabinovich et al., 2017)	85.9	87.1
coarse2fine (Dong and Lapata, 2018)	87.7	88.2
Our Neurals		
Bi-GRU	85.93	86.42
Transformer	**87.95**	86.78

There are two types of data set splits: question-split and query-split. In the former, training and testing examples are divided based on questions, thus based on the input sequence. In the latter, training and test examples are divided according to the similarity of their meaning representations, thus based on output sequences. In other words, training and testing examples in query-split are strictly controlled to have a more diverse set of logical formulae. Therefore, its use is more appropriate when evaluating the model's capability to compose output sequences (i.e., lambda calculus expressions here).

For each split, data sets might contain variables with or without anonymisation (Tables 1 and 2, resulting in two versions of the first data set (i.e., ATIS question-split and ATIS question-split non-anonymised) and three versions of the second data set (i.e., GEO question-split, GEO question-split non-anonymised, and GEO query-spit non-anonymised). Versions of GEO without anonymisation are from Kennardi et al. (2019) and splits originate from Finegan-Dollak et al. (2018).

As output logical formulae cannot be partially correct, we report the exact match by computing

$$\text{Accuracy} = \frac{\text{\# of correct formulae}}{\text{\# test examples in the test set}}.$$

4 Results

Our self-attention neural semantic parsing model became the new state-of-the-art on ATIS with

Table 5: The accuracy [%] on non-anononymised (NA) ATIS and GEO test sets

Model	ATIS NA	GEO NA quest split	GEO NA query split
Seq2Seq + Attention (Dong and Lapata, 2016)	72.02	67.39	41.94
coarse2fine (Dong and Lapata, 2018)	**79.1**	72.4	52.69
Bi-GRU	73.41	72.4	56.45
Transformer	75.99	**75.27**	**63.98**

Table 6: Difficult examples

Data set	Input	Output
ATIS	fare code fb0 what doe that mean	fb0
	what type of plane is a ac0	ac0
GEO	what is the average population per square km in s0	density:i s0
	what is the length of the r0 in s0	len:i r0

its accuracy of 87.95% (Table 4). However, the best result on GEO was by a statistical semantic parser called *Type-Driven Incremental Semantic Parsing* (TISP) (Zhao and Huang, 2015). The result was explained by our model overfitting on GEO that has fewer examples than ATIS, regardless of us using a smaller self-attention model on GEO that on ATIS (i.e., 8 vs. 16 heads). As expected, regardless of the model, results entity with anonymisation were always better than without (Table 5). On ATIS (GEO), this difference was approximately 10% (at least 15%). The GEO query-split task — with more diverse input and output instances — was harder than the GEO question-split task. Results indicated that our model is capable of capturing relationships by learning token attributes as opposed to only one-to-one mappings from a token in a sentence to a token in a logical formula.

Thus, Transformer was powerful in semantic parsing. The model outperformed its baselines on ATIS, GEO question split, and GEO query split with the best accuracy values of 87.95%, 75.27%, 63.98%, respectively. Our implementation of Bi-GRU was also competitive, achieving better results than the baseline model from (Dong and Lapata, 2016) across these data and outperforming all baselines on GEO query-split. We argued that Transformers are better at capturing long distance dependencies a the model process an sentence is process as a whole, instead of word by word. However, the Transformer implemented in this research is known to have an upper limit to the distances over it can easily learn relationships (Dai et al., 2019).

Token generation was an important feature in our comparisons although theoretically the difference between Seq2Seq's Long Short-Term Memory (LSTM) and our basic model's Bi-Directional Gated Recurrent Units (Bi-GRU, which performed substantially worse) should have been minor. Seq2Seq used a greedy search[1] for token generation while all other models beam searched,[2] which tends to be a better choice for sequence-to-sequence models.

Table 6 shows example instances that were difficult for every model. There was a considerable difference between the length of input sentences (Input column) and their corresponding logical forms (Output column). This was explained by sequence-to-sequence models' tendency to not choose the end of sequence (<eos>) when beginning the generation process, because of them having learnt that logical formulae are usually longer than one or two tokens (i.e., the probability of <eos> is low in the beginning of decoding which makes the mapping from long inputs to short outputs inaccurate).

5 Conclusion and Future Work

We evaluated the Transformer architecture for semantic parsing. The model was extensively evaluated with two data sets from different domains — with and without anonymisation — across a range of complexity levels. Experiments shows Transformer is competitive with other state-of-the-art models and outperformed strong baselines in some settings.

For future work, it would be interesting to design a tree-structure self-attention model. As logical forms are tree-structures, adding some constraints in the decoder to enforce tree-based decoding would be of particular interest.

[1]Greedy search generates the next token with the highest probability relating to the current output sequence. While this strategy is suitable for the current timestamp, it may be a sub-optimal choice to construct the full output formula.

[2]Beam search has k-best output sequences each time and it considers all options of combining those sequences and all candidates in the vocabulary. Then, it chooses k-best output sequences to generate the end of sequence.

Acknowledgement

We are thankful for our co-supervised student's contribution. Namely, we express our gratitude to Xiang Li for his insight throughout his Bachelor of Advanced Computing (Honours) project (Li, 2019) in the Australian National University in 2019 that founded this study. We also thank the Australasian Language Technology Association and anonymous referees of its 2020 workshop for their helpful comments.

References

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Marco Damonte, Rahul Goel, and Tagyoung Chung. 2019. Practical semantic parsing for spoken language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 16–23, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2016, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2018, pages 731–742, Melbourne, Australia. Association for Computational Linguistics.

Xing Fan, Emilio Monti, Lambert Mathias, and Markus Dreyer. 2017. Transfer learning for neural semantic parsing. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, Rep4NLP 2017, pages 48–56, Vancouver, Canada. Association for Computational Linguistics.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

Jonathan Herzig and Jonathan Berant. 2017. Neural semantic parsing over multiple knowledge-bases. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL 2017, pages 623–628, Vancouver, Canada. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2016a. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2016, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2016b. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Alvin Kennardi, Gabriela Ferraro, and Qing Wang. 2019. Domain adaptation for low-resource neural semantic parsing. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 87–93, Sydney, Australia. Australasian Language Technology Association.

Tomáš Kočiský, Gábor Melis, Edward Grefenstette, Chris Dyer, Wang Ling, Phil Blunsom, and Karl Moritz Hermann. 2016. Semantic parsing with semi-supervised sequential autoencoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2016, pages 1078–1087, Austin, TX, USA. Association for Computational Linguistics.

Xiang Li. 2019. *Improving Semantic Parsing with Self-Attention Model. Bachelor of Advanced Computing (Honours) Thesis*. The Australian National University, Canberra, ACT, Australia.

Wang Ling, Phil Blunsom, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Fumin Wang, and Andrew Senior. 2016. Latent predictor networks for code generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2016, pages 599–609, Berlin, Germany. Association for Computational Linguistics.

P. J. Price. 1990. Evaluation of spoken language systems: the ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Maxim Rabinovich, Mitchell Stern, and Dan Klein. 2017. Abstract syntax networks for code generation and semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1139–1149, Vancouver, Canada. Association for Computational Linguistics.

Raymond Hendy Susanto and Wei Lu. 2017. Neural architectures for multilingual semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL 2017, pages 38–44, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, 31(7):1235–1270.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the 13th National Conference on Artificial Intelligence*, volume 2, pages 1050–1055, Portland, Oregon, USA. AAAI Press / The MIT Press.

Luke Zettlemoyer and Michael Collins. 2007. Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 678–687, Prague, Czech Republic. Association for Computational Linguistics.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, pages 658–666, Arlington, Virginia, United States. AUAI Press.

Kai Zhao and Liang Huang. 2015. Type-driven incremental semantic parsing with polymorphism. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1416–1421, Denver, Colorado. Association for Computational Linguistics.

Overview of the 2020 ALTA Shared Task: Assess Human Behaviour

Diego Mollá
Department of Computing
Macquarie University
diego.molla-aliod@mq.edu.au

Abstract

The 2020 ALTA shared task is the 11th instance of a series of shared tasks organised by ALTA since 2010. The task is to classify texts posted in social media according to human judgements expressed in them. The data used for this task is a subset of SemEval 2018 AIT DISC, which has been annotated by domain experts for this task. In this paper we introduce the task, describe the data and present the results of participating systems.

1 Introduction

Human behaviour can be negatively or positively assessed based on a reference set of social norms. When judgement is explicitly stated in narratives, e.g., "They are hard-working and honest.", we can attempt to encounter appraisal words such as "hard-working" and "honest" used between interlocutors for advancing their judgement.

Attitude positioning plays an important role in Martin and White's (2005) Appraisal framework[1] (AF) for analysing someone's use of evaluative language to negotiate solidarity.

To the best of our knowledge, no prior work has attempted to automatically codify text using the AF judgement categories. The goal of the 2020 ALTA shared task is to develop a computational model that can identify and classify judgements expressed in textual segments. Participants are challenged to predict the judgement appraised by classifying each short-text message into one or more label candidates (or none): *normality, capacity, tenacity, veracity, propriety*.

2 The 2020 ALTA Shared Task

The 2020 ALTA Shared Task is the 11th of the shared tasks organised by the Australasian Language Technology Association (ALTA). As in previous shared tasks, it targets university students with programming experience, but it is also open to graduates and professionals. The general objective of these shared tasks is to introduce interested people to the sort of problems that are the subject of active research in a field of natural language processing. Depending on the availability of data, the tasks have ranged from classic but challenging tasks to tasks linked to very hot topics of research. Details of the 2020 ALTA Shared task and past tasks can be found in the 2020 ALTA Shared Task website.[2]

There are no limitations on the size of the teams or the means that they may use to solve the problem. We provide training data but participants are free to use additional data and resources. The only constraint in the approach is that the processing must be fully automatic — there should be no human intervention.

As in past ALTA shared tasks, there are two categories: a student category and an open category.

- All the members of teams from the **student category** must be university students. The teams cannot have members that are full-time employed or that have completed a PhD.
- Any other teams fall into the **open category**.

The prize is awarded to the team that performs best on the private test set — a subset of the evaluation data for which participant scores are only revealed at the end of the evaluation period.

3 The Appraisal Framework

The Appraisal framework (AF) is concerned with the use of linguistic markers for identifying and track the ways attitudes are invoked in authored

[1] https://www.grammatics.com/appraisal/

[2] http://www.alta.asn.au/events/sharedtask2020/

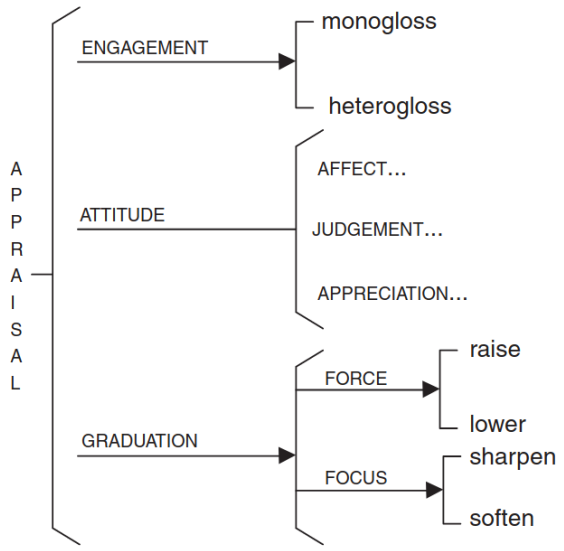


Figure 1: Overview of appraisal resources (Martin and White, 2005, p38)

text. The framework defines three subsystems for evaluative meaning making (1) ATTITUDE; (2) ENGAGEMENT; and (3) GRADUATION. Each of these are further divided in to other subsystems (Figure 1). In particular, The ATTITUDE framework is divided into three subsystems: (1) AFFECT (registering of emotions); (2) APPRECIATION (evaluations of natural and semiotic phenomena); and (3) JUDGEMENT (evaluations of people and their behaviour).

The judgement subsystem has two regions: social esteem and social sanction. The subcategories of each of these two regions form the target labels for the 2020 ALTA Shared Task. In particular:

Social esteem tends to function as admiration or criticism and can be subdivided into three subcategories:

Normality (how unusual one is): "He is old-fashioned".

Capacity (how capable one is): "Self-driven 12 year old is a maths genius".

Tenacity (how resolute one is): "They are hard-working and honest".

Social sanction functions as praise or condemnation and can be subdivided into two subcategories:

Veracity (how honest/truthful one is): "They are hard-working and honest".

Propriety (how ethical one is): "She is too arrogant to learn the error of her ways".

The judgement system is used to assess human behaviour and their position on certain social norms. Further details and examples can be found in The Appraisal Website.[3]

4 Data

The source data of the 2020 ALTA Shared Task is a subset of the SemEval 2018 AIT DISC dataset.[4] A total of 300 tweets have been manually annotated in a two-stage process. The annotation was first annotated by two linguists from two Australian universities (University of Wollongong and University of New South Wales) and then double-checked by two other linguists from the same two universities. The data were subsequently split into a training set of 200 tweets, and a test set of 100 tweets.

Each tweet was annotated with one or more (or none) of the following labels: *normality, capacity, tenacity, veracity, propriety*. Table 1 shows artificial examples of text messages and their annotations.

5 Evaluation

As in previous ALTA shared tasks, the task was managed as a Kaggle in Class competition. This year's task name was "ALTA 2020 Challenge".[5] The Kaggle-in-Class platform enabled the participants to download the data, submit their runs, and observe the results of their submissions in a leaderboard instantly.

As is common in Kaggle competitions, when a participant team submits their results, the public leaderboard shows the evaluation results of part of the test data, and the results of the remaining test data are held for the final ranking. By following the public leaderboard, a team can then gauge the performance of their system in comparison with that of other systems in the same public test set. A team can choose up to two of their runs for the final ranking. If a team chooses runs for the final ranking, the best results on these runs on the private partition of the test data will be used. If a team

[3] https://www.grammatics.com/appraisal/appraisalguide/unframed/stage2-attitude-judgement.htm

[4] https://competitions.codalab.org/competitions/17751#learn_the_details-datasets

[5] https://www.kaggle.com/c/alta-2020-challenge/

Text	Normality	Capacity	Tenacity	Veracity	Propriety
Read and try to comprehend what you have commented on.	0	1	0	0	0
Fans of adoring Dictatorships and Totalitarians.	0	0	0	0	1
Keep going like you always have done.	0	0	1	0	0
She showed her true colors.	0	0	0	0	1
He is a nasty person.	1	0	0	0	1
Corruption 101	0	0	0	1	0

Table 1: Artificial examples of texts and their annotations.

does not choose any runs, the private evaluation results of the run with the best results on the public partition will be chosen.

The systems were evaluated using the mean of the F1 score over the test samples (1),

$$\begin{aligned} F1 &:= \frac{1}{|S|}\sum_{s\in S} F_\beta(y_s,\hat{y}_s) \\ F_1(y_s,\hat{y}_s) &:= 2\frac{P(y_s,\hat{y}_s)\times R(y_s,\hat{y}_s)}{P(y_s,\hat{y}_s)+R(y_s,\hat{y}_s)} \\ P(y_s,\hat{y}_s) &:= \frac{|y_s\cap\hat{y}_s|}{|y_s|} \\ R(y_s,\hat{y}_s) &:= \frac{|y_s\cap\hat{y}_s|}{|\hat{y}_s|} \end{aligned} \quad (1)$$

where y_s is the set of predicted labels in sample s, $\hat{y}_s$) is the set of true labels in the sample, and S is the set of samples. If there were no true or no predicted labels, $F_1(y_s,\hat{y}_s) := 0$.

6 Participating Systems

In total 5 teams registered for the competitions, all of them in the student category. Of these, 3 teams submitted runs.

Team NLP-CIC experimented with logistic regression and Roberta (Aroyehun and Gelbukh, 2020). Whereas the logistic regression classifier obtained the best results in the public leaderboard, it performed much worse in the private leaderboard. In contrast, the Roberta classifier obtained consistent results in both the public and private leaderboards.

Team OrangutanV2 designed classifiers using ALBERT and transfer learning (Parameswaran et al., 2020). After observing that 22 tweets from the test set are also in the training set, they also incorporated a component that performed cosine similarity with the samples from the training data.

Team NITS experimented with ensemble approaches (Khilji et al., 2020). They obtained pretrained word embeddings and incorporated polynomial features. These features were fed to decision tree and Extreme Gradient Boosting (XGBoost) classifiers.

7 Results

Table 2 shows the results of the systems in the private leaderboard.

Team	F1	p
NLP-CIC	0.155	
OrangutanV2	0.105	0.313
NITS	0.053	0.010

Table 2: Results of the participating teams according to the private leaderboard. Column p indicates the Wilcoxon Signed Rank test between a team and the top team after removing ties.

The results indicate that this task has been particularly challenging and there is room for improvement. A possible reason for the difficulty of this task is the small number (200) of annotated samples available. Another reason for the low results is the relatively large percentage of samples with empty judgements. In particular, 60% of the test data had empty judgements. According to Formula (1), the F1 score of test samples with no annotations is 0. This means that the upper bound with this test data is 0.4.

8 Conclusions

The aim of the 2020 ALTA shared task was to predict the judgement of short texts according to Martin and White's (2005) Appraisal framework. The task proved challenging, presumably due to the small amount of annotated data and the sparse annotations in the data.

Acknowledgments

We thank the anonymous sponsor who donated the data for this shared task.

References

Segun Taofeek Aroyehun and Alexander Gelbukh. 2020. Automatically predicting judgement dimensions of human behaviour. In *Proceedings of the 18th Annual Workshop of the Australasian Language Technology Association*.

Abdullah Faiz Ur Rahman Khilji, Rituparna Khaund, and Utkarsh Sinha. 2020. Human behavior assessment using ensemble models. In *Proceedings of the 18th Annual Workshop of the Australasian Language Technology Association*.

J. Martin and P. White. 2005. *The Language of Evaluation Appraisal in English*. Palgrave Macmillan, UK.

Pradeesh Parameswaran, Andrew Trotman, Veronica Liesaputra, and David Eyers. 2020. Classifying JUDGEMENTS using transfer learning. In *Proceedings of the 18th Annual Workshop of the Australasian Language Technology Association*.

Automatically Predicting Judgement Dimensions of Human Behaviour

Segun Taofeek Aroyehun
CIC, Instituto Politécnico Nacional
Mexico City, Mexico
aroyehun.segun@gmail.com

Alexander Gelbukh
CIC, Instituto Politécnico Nacional
Mexico City, Mexico
www.gelbukh.com

Abstract

This paper describes our submission to the ALTA-2020 shared task on assessing behaviour from short text, We evaluate the effectiveness of traditional machine learning and recent transformers pre-trained models. Our submission with the Roberta-large model and prediction threshold achieved first place on the private leaderboard.

1 Introduction

Language enables us to express evaluation of people, action, event, and things. This manifests as emotion and assessment of human behaviour and artefacts. The study of evaluative language has benefited from efforts in several disciplines such as linguistics, philosophy, psychology, cognitive science and computer science (Benamara et al., 2017). In linguistics, the appraisal framework of Martin and White (2003) provides a detailed classification scheme for understanding how evaluation is expressed and implied in language. In computer science, affective computing study evaluative language under the umbrella term of sentiment analysis with common tasks involving detection and classification of polarity and emotion, and aspect-based sentiment analysis, among others. Sentiment analysis has benefited from the availability of user-generated content on online platforms.

The theory of appraisal proposed by Martin and White (2003) has three categories of evaluative text: affect, judgement, and appreciation. These categories respectively model opinions in terms of emotions, norms, and aesthetics. Utterances are viewed as indicating positive ("praising") or negative ("blaming") disposition towards some object (person, thing, action, situation, or event). The judgement dimensions are normality, capacity, tenacity, veracity, and propriety. Each of the dimensions represents an answer to the following corresponding questions:

- Normality: How special?
- Tenacity: How dependable?
- Capacity: How capable?
- Veracity: How honest?
- Propriety: How far beyond reproach?

The corpus used in this paper is annotated with the above judgement dimensions.

Taboada and Grieve (2004) automatically categorized appraisal into affect, judgement, and appreciation using a lexical approach that groups adjectives according to their semantic orientation. Benamara et al. (2017) surveyed linguistic and computational approaches to the study of evaluative text. Their analysis suggested that appraisal is a richer and more detailed task amenable to computational approaches subject to availability of data. They envision that appraisal analysis can contribute to the advances in affective computing. Recently, Hofmann et al. (2020) showed that dimensions of appraisal can improve emotion detection in text. A similar observation was made by Whitelaw et al. (2005) who found appraisal phrases as useful features for sentiment analysis.

This paper investigates the capabilities of machine learning models in predicting the dimensions of human judgement expressed in short texts (tweets) as part of the ALTA-2020 shared task on assessing human behaviour (Mollá, 2020). The task aims to advance computational techniques for analysing evaluative language.

The use of neural networks has lead to significant performance improvements in NLP tasks. However, neural networks require a large amount of labeled data. On the contrary, the traditional machine learning models such as NBSVM are competitive in low-data regimes (Wang and Manning, 2012; Aroyehun

Label	Normality	Capacity	Tenacity	Veracity	Propriety
Proportion	0.11	0.16	0.11	0.015	0.18

Table 1: Frequency of each label in the training set as a fraction of the total number of examples.

and Gelbukh, 2018). The recently introduced contextual representation learning models (Peters et al., 2018; Devlin et al., 2019) are pre-trained with language modeling objective on a large and diverse collection of text. The learned representation can be transferred to downstream tasks via fine tuning (Howard and Ruder, 2018). We examine the effectiveness of using NBSVM and fine tuning a Roberta-large model (Liu et al., 2019) for predicting dimensions of judgement expressed in short text.

2 Methodology

Task. Given a short text predict one or more judgement dimensions expressed in the given text. This is a multilabel classification problem where the labels consist of the five judgement dimensions.

Data. We employed the data provided by the organizers of the ALTA-2020 shared task (Mollá, 2020). The training set has 198 tweets. Each example is annotated with the presence or absence of each of the judgement dimensions as outlined in Section 1. Table 1 shows the proportion of each label in the training set. The proportion ranges from 2% to 18%. The test set consists of 100 examples. About 50% each is used for the public and private leaderboards for the competition on Kaggle[1] In-class platform.

The private leaderboard is used for the final ranking, the scores are available after the completion of the competition while the public leaderboard is used by the competition participants to evaluate their models during the competition. In our experiment using the Roberta-large model, we created a validation set by randomly sampling 10% of the training set.

Data Pre-processing. We clean the text of each tweet by removing punctuation marks, digits, and repeated characters. We normalize URLS and usernames (tokens that starts with the @ symbol). Hashtags are converted to their constituent word(s) after removing the # symbol.

NBSVM. Wang and Manning (2012) proposed a support vector machine (SVM) model that uses the naive bayes log-count ratio as features. NBSVM is a strong linear model for text classification. In our implementation we use the logistic regression classifier in place of the SVM. The features are based on word n-grams (unigrams and bigrams). We experiment with and without the data pre-processing step. In the multi-label classification setting, we train a binary classifier per label with the same classifier settings.

Roberta-large. An optimized BERT (Devlin et al., 2019) model trained for longer and on larger and more diverse text collection totalling 160GB. In addition, the pre-training tasks did not include next sentence prediction and the tokenizer is based on BPE (Liu et al., 2019). We fine tune the model on the data provided by the task organizers without the data pre-processing step. We used the simpletransformers library [2] for our experiment. The classifier is a linear layer with sigmoid activation function. The hyperparameters are: maximum learning rate of $4e-5$, number of epochs is 20 with early stopping on the validation loss using a patience of 3, batch size of 64, the model parameters are optimized using AdamW with a linear schedule and a warm up steps of 4 and the maximum sequence length is 128.

Prediction threshold. Lipton et al. (2014) studied the difficulty of relating the maximum achievable F1 score with the decision thresholds on predicted conditional probabilities. They observed that selecting predictions that maximize the F1 score is a function of the conditional probability assigned to an example and the distribution of conditional probabilities for other examples. Following this observation, we choose decision threshold for each label to track the distribution of conditional probabilities on the validation set without reference to the gold labels, to avoid overfitting. The default decision threshold is 0.5 and we find that the conditional probabilities are significantly less. We apply this heuristic to the model outputs of the Roberta-large model. Specifically, we set 0.2 as the decision

[1] https://www.kaggle.com/

[2] https://simpletransformers.ai/

Method	Public leaderboard	Private leaderboard	Average
NBSVM	**0.16000**	0.00000	0.08000
NBSVM w/ prep.	0.16000	0.00000	0.08000
Roberta-large	0.11666	0.06666	0.09166
Roberta-large w/ threshold	0.14285	**0.15466**	**0.14876**

Table 2: Mean F1 score on the public and private test sets. Average is the unweighted mean of the scores on the private and public leaderboards as they are approximately 50% each of the test set.

threshold for the *capacity* label and 0.1 for the remaining labels.

3 Results

Table 2 shows the results obtained on the test set split into two equal halves as the public and private leaderboards. With the NBSVM model, we achieved the best score of 0.16 on the public leaderboard. The application of data pre-processing step did not impact the performance of the NBSVM model, probably because the tokens removed are not relevant lexical units for the task. Following this observation, we did not apply the pre-processing step to our experiments with the Roberta-large model. The Roberta-large model obtained a relatively lower score on the public leaderboard and appears to generalize better on the other half of the test set as shown by the scores on the private leaderboard. There is a significant performance improvement due to the decision thresholding on the Roberta-large model outputs. With this strategy, we achieved the best overall score on the ALTA-2020 competition.

4 Conclusion

We address the task of automatically predicting judgement dimensions in the context of the ALTA-2020 shared task. We evaluated the performance of a strong linear classifier, NBSVM with n-grams as features and a recent pre-trained language model, Roberta-large. We observed that the NBSVM achieves our best score on the public leaderboard but it did not generalize to the private test set. The Roberta-large model with decision thresholding strategy showed consistent performance on both the public and private leaderboards. With this model, we achieved the best overall score on the competition.

While we achieved better performance with the Roberta-large model, we think that the statistical power (Card et al., 2020) of the test set is limited due to the small sample size (100 examples). As such, it is difficult to differentiate performance improvement by chance from substantial model advantage. We hope to test our approaches on a larger test set in order to examine the robustness of our approaches.

Acknowledgments

The authors thank CONACYT for the computer resources provided through the INAOE Supercomputing Laboratory's Deep Learning Platform for Language Technologies.

References

Segun Taofeek Aroyehun and Alexander Gelbukh. 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97.

Farah Benamara, Maite Taboada, and Yannick Mathieu. 2017. Evaluative language beyond bags of words: Linguistic insights and computational applications. *Computational Linguistics*, 43(1):201–264.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jan Hofmann, Enrica Troiano, Kai Sassenberg, and Roman Klinger. 2020. Appraisal theories for emotion classification in text. *arXiv preprint arXiv:2003.14155*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. 2014. Optimal thresholding of classifiers to maximize f1 measure. In *Machine Learning and Knowledge Discovery in Databases*, pages 225–239, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

James R. Martin and Peter R. White. 2003. *The language of evaluation*, volume 2. Springer.

Diego Mollá. 2020. Overview of the 2020 ALTA Shared Task: Assess Human Behaviour. In *Proceedings of the 18th Annual Workshop of the Australasian Language Technology Association*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Maite Taboada and Jack Grieve. 2004. Analyzing appraisal automatically. In *In Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*.

Sida I. Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94.

Casey Whitelaw, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM International Conference on Information and knowledge management*, pages 625–631.

Classifying JUDGEMENTS using Transfer Learning

Pradeesh Parameswaran, Andrew Trotman, Veronica Liesaputra, David Eyers
Department of Computer Science
University of Otago
New Zealand
[pradeesh,andrew,veronica,dme]@cs.otago.ac.nz

Abstract

We describe our method for classifying short texts into the APPRAISAL framework, work we conducted as part of the ALTA 2020 shared task. We tackled this problem using transfer learning. Our team, "*orangutanV2*" placed equal first in the shared task, with a mean $F1$-score of 0.1026 on the private data set.

1 Introduction

Systemic Functional Linguistics (SFL) is a theory of language which examines the relationship between language meaning and the functions in a social context (Halliday, 1996). One popular framework that uses SFL is APPRAISAL (Martin and White, 2005). The APPRAISAL framework is based on the notion of uncovering the attitude of the author from the perspective of a potential listener or reader. It is used by linguists in analysing human behaviour from textual data (Ross and Caldwell, 2020; Starfield et al., 2015; Wu, 2013; Hommerberg and Don, 2015). Figure 1 shows an overview of the APPRAISAL framework.

The three main resources of the APPRAISAL framework are; ATTITUDE, ENGAGEMENT and GRADUATION (Martin and White, 2005). The ATTITUDE framework is then subdivided into three subsystems; AFFECT (emotions), APPRECIATION (evaluation of natural and semiotic phenomena) and JUDGEMENT (evaluation of people and their behaviour). The JUDGEMENT subsystem can be divided into two categories: SOCIAL ESTEEM and SOCIAL SANCTIONS. SOCIAL ESTEEM primarily involves admiration and criticism and SOCIAL SANCTION involves praise and condemnation.

SOCIAL SANCTIONS can be further divided into three subcategories: *normality* (how usual one is), *capacity* (how capable one person is) and *tenacity* (how dependable one is). As for SOCIAL SANCTION it can be further divided into two subcategories; *veracity* (how truthful one is) and *propriety* (how virtuous one is).

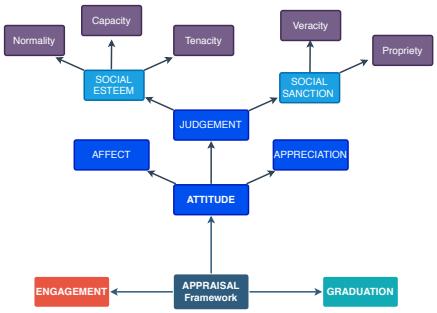


Figure 1: The APPRAISAL Framework (Adapted From (Martin and White, 2005))

The robustness of the APPRAISAL framework lies in its ability to be used in various different social contexts. It also offers linguists detailed strategies for realising the framework (Ngo and Unsworth, 2015). Since its debut, the APPRAISAL framework has been widely used to explore how language is being used in various different environments such as in analysing examiners' reports on doctoral theses (Starfield et al., 2015), Donald Trump's rhetoric tweets (Ross and Caldwell, 2020), people's perception on the outcome of the Brexit referendum (Bouko and Garcia, 2020) and in teaching English as a second language (ESL) (Ngo and Unsworth, 2015).

Currently, linguists manually classify sentences using annotation software as there is no automated classification technique that exists to automate the task (Fuoli, 2018). Thus, this problem sparked the interest of Australasian Technology Association (ALTA) to organise a shared challenge task to develop a model that can automatically identify and classify human behaviour (JUDGEMENT) ex-

Feature matched	Number
(None)	104
Normality	22
Capacity	31
Tenacity	21
Veracity	2
Propriety	33
Multiple Features	74

Table 1: Distribution and Pattern of Training Data

pressed in tweets on Twitter (Molla, 2020). The task was to classify tweets into either one or more (or none) of the five sub-categories of JUDGEMENT.

We present our participation in this challenge. We tackled this problem by utilising a pre-trained transfer learning model, ALBERT (Lan et al., 2019), as a classifier.

2 Data Set

The data set[1] provided by the organisers is a collection of 300 tweets from SemEval 2018 (Mohammad et al., 2018): 200 tweets for training and 100 for testing. The training data set consists of tweet ID and the labelled annotations of sub-categories of JUDGEMENT which the tweet belongs to. If the tweet does not contain any sub-categories it is marked as blank.

We analysed the training data to understand the distribution and the patterns of category use. Table 1 describes the pattern. The data set is not balanced between categories, particularly for *Veracity* where there are only have 2 examples in the training set. We have also found that there is 1 duplicate tweet in the training data and we promptly informed the organisers of this. Additionally we found 22 of the tweets in training data are in the testing data.

3 Methodology

First we handle class imbalances followed by pre-processing of our tweets. Then, we perform unsupervised classification by utilising ALBERT's pre-trained model. We chose ALBERT because it performs reasonably well on various different tasks such as offensive language detection (Zampieri et al., 2020), multiple-choice reading comprehension (Si et al., 2019), and question and answering (Khashabi et al., 2020). Finally, we employed the cosine similarity measure in order to correct the mistakes made by our classifier.

We have made our system's source code publicly available on Github.[2]

3.1 Handling Class Imbalance

Due to a low number of training examples for *Veracity*, we removed this category from our examples (and, consequently, results). Early experiments showed that this led to a significant performance improvement. We did not make any adjustments to any other categories.

3.2 Pre-Processing Data

We experimented with various pre-processing strategies of including stemming, removing mentions, hashtags and URLs. From our early experiments, we found that by removing mentions from the tweets, and keeping the text as is, yielded the best performance.

3.3 ALBERT Transfer Learning Classifiers

We used huggingface's[3] implementation of ALBERT. We then added a sigmoid classifier (for binary classification) or softmax classifier (for multi-label classification) on top of the model to predict the probability of a category. We built three separate classifiers using this model; a binary classifier for SOCIAL SANCTIONS (C_s), a binary classifier for SOCIAL ESTEEM (C_e) and a multi-label classifier to classify the potential categories the tweet belongs to (C_m).

First we feed our pre-processed data into C_s and C_e. Once the texts get classified to be either both or one of the categories, we continue to feed it to C_m in order to get the potential granular categories.

We evaluated the performance of our classifier by splitting our training data into 70% for training, 10% for validation and 20% for evaluation purposes. We used the Adam Optimizer with a learning rate of 10^{-5} for 50 epochs. We set the batch size to be 64. We used our validation set's mean $F1$ score as an early stopping criterion. We stop the training if the score does not increase for 15 consecutive epochs, or the maximum number of epochs has been reached.

[1] https://www.kaggle.com/c/alta-2020-challenge/data

[2] https://github.com/prasys/alta2020-appraisal

[3] https://huggingface.co/

Figure 2 shows the recall and precision scores of the three different classifiers on our evaluation set. We set the class probabilities confidence level to be 0.5 in order to maximise precision and recall scores. Our C_s classifier in Figure 2a obtained both recall and precision score of 73.68% and C_e classifier in Figure 2b obtained both recall and precision score of 93.71%. As for our multi-label classification in Figure 2c we obtained a precision score of 56.25% and a recall 42.86%.

From visual inspection of the training data and our result, we observed that the C_m classifier can be further improved by adding personal pronoun detection of third person pronouns. We encoded this feature as a binary value. We used Google Natural Language Processing API[4] to extract the pronouns. Then we append the values in the final layer of our model before the softmax classifier.

3.4 Document Cosine Similarity

In the test set we observed the presence of 22 prelabeled tweets from the training set. To correct classifier mistakes, we used the Universal Sentence Encoding (Cer et al., 2018) to perform cosine similarity between the training data and the test data. Our solution was generic. We converted tweets into a high dimension vector representation, computed the cosine similarity with the training data, and those above a given threshold were considered to be the correct answer. We set the threshold to 1 in order to catch only exact matches.

4 Results

Kaggle was used as the platform for run submission. In Kaggle, the test data provided to us by the ALTA organisers is split into public (public leaderboard) and private (private leaderboard). The private portion serves as a validation portion in order for the organisers to determine the effectiveness of all systems. The scores are evaluated using mean $F1$. We summarise and present our results in Table 2.

4.1 Discussion

The objective set by the organisers of the shared task at ALTA was to create a baseline for this task.

(a) C_s Classifier

(b) C_e Classifier

(c) C_m Classifier

Figure 2: Recall and Precision Scores of C_s, C_e & C_m Classifiers

System	Public Score	Private Score
C_m (Baseline)	0.19333	0.06133
C_m + Cos. Similarity	0.20333	0.08133
C_s + C_e + C_m + Cos. Similarity	**0.21333**	0.08133
C_s + C_e + C_m + Cos. Similarity + Pronoun	0.20000	**0.10266**

Table 2: System Evaluation

[4] https://cloud.google.com/natural-language/docs

Although our system placed 2nd on both private and public leader boards, statistical tests (run by the challenge organisers) showed no statistically significant difference between the scores of our team and the team that got a slightly higher score. Both were declared joint winners.

Our further investigations suggest that our system performed well at identifying SOCIAL SANCTIONS, probably because the important words in the tweets appear close to each other in the vector space.

Equally, we are not performing well at classifying SOCIAL ESTEEM. Although, our binary classifier is able to classify tweets belonging to SOCIAL ESTEEM with a high degree of accuracy, we are not able to classify them accurately at a subcategorical level. This prompted us to look deeper into the problem and to offer several ways to improve this task – which we discuss in subsections 4.2 and 4.3.

4.2 Lack of Training Data

The primary difficulty in achieving higher accuracy in classifying tweets is the limited amount of training data available (Lu et al., 2014). Whilst acknowledging the fact that annotating a large set of data manually is challenging (Ciravegna et al., 2002), we propose that a smaller data set such as the one being used for this task should be tailored to be a specific topic rather than being spread across multiple topics. For instance, if the topic were the recent New Zealand elections, we may be able to improve the performance of the classifier by augmenting it with domain knowledge obtain from news sources or the Wikipedia (Yangarber et al., 2000; Gabrilovich and Markovitch, 2006). This is similar to how humans used domain knowledge to resolve ambiguity in evaluating the APPRAISAL framework with Trump's tweets (Ross and Caldwell, 2020).

4.3 The Annotation Process

Identifying expression of APPRAISAL in a piece of text is not as straightforward as some discourse analysis tasks (Mauranen and Bondi, 2003). Although Martin and White (2005) discuss the framework in detail and provides examples, there is the potential for the "Russian doll syndrome" (Thompson, 2014), where classifying into one category can be interpreted as indirectly classifying into other categories. This creates a problem in providing a reliable annotation. We show two examples from the training data set, with the provided categories in bold—

> "*@Gennneral thanks gen!! Love you miss you happy birthday natong duha* ❤️ ❤️ ❤️." ("**None of the above**")

> "*@priny_baby happppy happpppyyyyy happpppppyyyyy birthday best friend!! Love you lots #chapter22* ❤️ ❤️ ❤️." ("**Normality**")

In the first example, the annotators classified it as being none of the 5 categories, whereas in the second example this was not the case. In both cases, our system predicted *Normality*. Our deep learning model was not able to accurately distinguish between these two tweets. We hypothesise that humans face a similar difficulty with these two tweets, and may not choose deterministically. One way of addressing ambiguity is to follow Fuoli (2018) and to use a step-wise method to ensure reproducibility of annotations. We plan to explore this further as part of our future work.

5 Conclusion and Future Work

We presented our approach in order to automatically identify and classify JUDGEMENT expressed in textual segments. We competed in the ALTA 2020 challenge under the team name of "*orangutanV2*" and placed equal first. Our best-performing system used a combination of transfer learning and document cosine similarity.

Despite setting a baseline for future work, we believe that there is still much work to be done in this area. As part of future work we are planning to tackle this problem in several ways, including:

- Looking at human level performance; and
- Experimenting with various different transfer learning models.

Acknowledgements

We would like to thank Google Cloud for their support in providing the infrastructure to conduct our experiments. We would also like to thank the ALTA organisers (especially Dr. Diego Molla-Aliod) for organising the challenge, and promptly replying to our enquiries.

References

Catherine Bouko and David Garcia. 2020. Patterns of Emotional Tweets: The Case of Brexit After the Referendum Results. In *Twitter, the Public Sphere, and the Chaos of Online Deliberation*, pages 175–203.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, Yun Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings*.

Fabio Ciravegna, Alexiei Dingli, Daniela Petrelli, and Yorick Wilks. 2002. User-system cooperation in document annotation based on information extraction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2473(October):122–137.

Matteo Fuoli. 2018. A stepwise method for annotating appraisal. *Functions of Language*, 25(2):229–258.

Evgeniy Gabrilovich and Shaul Markovitch. 2006. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the National Conference on Artificial Intelligence*.

Michael AK Halliday. 1996. Literacy and linguistics: A functional perspective. In *Literacy in society 339*, page 376.

Charlotte Hommerberg and Alexanne Don. 2015. Appraisal and the language of wine appreciation: A critical discussion of the potential of the Appraisal framework as a tool to analyse specialised genres. *Functions of Language*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UnifiedQA: Crossing Format Boundaries With a Single QA System. In *arXiv preprint arXiv:2005.00700*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *arXiv preprint arXiv:1909.11942*, pages 1–17.

Zhongqi Lu, Yin Zhu, Sinno Jialin Pan, Evan Wei Xiang, Yujing Wang, and Qiang Yang. 2014. Source free transfer learning for text classification. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

J. R. Martin and P. R.R. White. 2005. *The Language of Evaluation*.

Anna Mauranen and M. Bondi. 2003. Evaluative language use in academic discourse. *Journal of English for Academic Purposes*.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 Task 1: Affect in Tweets. In *In Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Diego Molla. 2020. Overview of the 2020 alta shared task: Assess human behaviour. In *Proceedings of the 18th Annual Workshop of the Australasian Language Technology Association.*

Thu Ngo and Len Unsworth. 2015. Reworking the appraisal framework in ESL research: refining attitude resources. *Functional Linguistics*, 2(1):1–24.

Andrew S. Ross and David Caldwell. 2020. 'Going negative': An APPRAISAL analysis of the rhetoric of Donald Trump on Twitter. *Language and Communication*, 70:13–27.

Chenglei Si, Shuohang Wang, Min-Yen Kan, and Jing Jiang. 2019. What does BERT Learn from Multiple-Choice Reading Comprehension Datasets? *arXiv preprint arXiv:1910.12391*.

Sue Starfield, Brian Paltridge, Robert McMurtrie, Allyson Holbrook, Sid Bourke, Hedy Fairbairn, Margaret Kiley, and Terry Lovat. 2015. Understanding the language of evaluation in examiners' reports on doctoral theses. *Linguistics and Education*, 31:130–144.

Geoff Thompson. 2014. Affect and emotion, target-value mismatches, and Russian dolls. In *Evaluation in context*, volume 47, page 66. John Benjamins Amsterdam.

Hai-bin Wu. 2013. Appraisal Perspective on Attitudinal Analysis of Public Service Advertising Discourse. *English Language and Literature Studies*.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for Information Extraction. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *arXiv preprint arXiv:2006.07235*.