University of Sheffield

Using Pre-trained Language Models for Toxic Comment Classification

Zhixue Zhao

Supervisor: Dr Ziqi Zhang and Dr Frank Hopfgartner

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Information School

November 7, 2022

Declaration

I, Zhixue Zhao, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

To my loving mother, father and brother.

Abstract

Toxic comment classification is a core natural language processing task for combating online toxic comments. It follows the supervised learning paradigm which requires labelled data for the training. A large amount of high-quality training data is empirically beneficial to the model performance. Transferring a pre-trained language model (PLM) to a downstream model allows the downstream model to access more data without creating new labelled data. Despite the increasing research on PLMs in NLP tasks, there remains a fundamental lack of understanding in applying PLMs to toxic comment classification. This work focuses on this area from three perspectives.

First, we investigate different transferring strategies for toxic comment classification tasks. We highlight the importance of efficiency during the transfer. The transferring efficiency seeks a reasonable requirement of computational resources and a comparable model performance at the same time. Thus, we explore the continued pre-training in-domain which further pre-trains a PLM with in-domain corpus. We compare different PLMs and different settings for the continued pre-training in-domain.

Second, we investigate the limitations of PLMs for toxic comment classification. Taking the most popular PLM, BERT, as the representative model for our study, we focus on studying the identity term bias (i.e. prediction bias towards comments with identity terms, such as "Muslim" and "Black"). To investigate the bias, we conduct both quantitative and qualitative analyses and study the model explanations. We also propose a hypothesis that builds on the potential relationship between the identity term bias and the subjectivity of comments.

Third, building on the hypothesis, we propose a novel BERT-based model to mitigate the identity term bias. Our method is different from previous methods that try to suppress the model's attention to identity terms. To do so, we insert the subjectivity into the model along with the suggestion of the presence of identity terms. Our method shows consistent improvements on a range of different toxic comment classification tasks.

Acknowledgements

I am grateful to the Information School, the University of Sheffield, for providing the three-years Graduate Teaching Assistant Scholarship which made this work possible. I would not have been able to carry out my research without this support.

The journey of PhD under COVID-19 was bittersweet and inevitably lonely sometimes. Fortunately, there were so many people there helping and supporting me. Without them, I would have not made it so far. First and foremost, I would like to express my sincere gratitude to my supervisors who have supported and helped me over the last three and half years of research. Thanks to Ziqi for his invaluable advice and patience. Thanks to Ziqi for holding me back when I was too over-ambitious, for encouraging me to achieve new highs and for his thoughtful feedback on my work throughout this thesis. I also give huge thanks to Frank for taking on supervision of my thesis part-way through, for helping me develop my research focus, and for supporting me on the way to pursuing an academic career.

I would also like to thank Steph, Andy and other colleagues in the Information School. They were always warm, responsive and supportive. I am grateful for the excellent environment they have created for our PhD students.

I also extend my thanks to Omaima, my lovely desk-mate and lunch buddy. Thanks for her company and support through the COVID-19 pandemic. The thanks also go to Ziming, Xuanning, Natalie, Tany, Yida and many friends. Thanks for their invaluable company and support.

Finally, my appreciation goes out to my family and friends for their encouragement and support throughout my studies. In particular, thanks go to my mother for encouraging and supporting my passion for research, even though I choose to live so far away from her. Her understanding, support and selfless love mean a lot to me. Thanks to my brother for taking good care of my parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As user-generated contents thrive, so does the spread of toxic comment online. There is a growing body of literature that recognises the importance of detecting toxic comment and studies this problem from different perspectives. One main area of studies in online toxic comments is using machine learning methods to automatically detect toxic comments. This thesis belongs to this category, and investigates the problem by focusing on using the more recent state-of-the-art (SOTA) machine learning models based on pre-trained language models. This chapter provides a background of this thesis, and sets the stage for the subsequent chapters.

The remainder of this chapter is structured as follows. Section 1.1 explains the motivation of this thesis, particularly why we study the pre-trained language models for detecting toxic comments. Section 1.2 presents the three research objectives that guide this thesis. Section 1.3 discusses the related ethical concerns. Section 1.4 outlines the structure of this thesis. Last, Section 1.5 lists publications published during the thesis.

1.1 Motivation

The rise of multiple social media platforms have revolutionised the way how we communicate with and express ourselves to other people. Unfortunately, they have also led to an increasing number of negative behaviours online. Such behaviours are diverse, but include hate speech, cyberbullying, threats and abusive language (Waseem et al., 2017).

Hate speech is commonly defined as any form of communication that disparages individuals or groups on the basis of a personal characteristic such as race,

ethnicity, gender, sexual orientation, nationality or religion (Schmidt and Wiegand, 2017). Cyberbullying refers to online behaviours intended to hurt someone. These include disseminating negative, harmful, false, or mean content about a person (Committee, 2019). It can also include sharing personal or private information about someone intending to cause embarrassment or humiliation (Committee, 2019). Undoubtedly, there are many other types of negative user generated content (UGC) online. Several recent pieces of literature used the broad umbrella term of "toxic comment" to generally refer to these different types of unhealthy and negative UGC.

Unfortunately, toxic comments such as *"kill yrslef a$$hole"* or *"they should all burn in hell for what they've done"* are not uncommon online (Nobata et al., 2016). Duggan (2014) have shown that 73% of adult internet users have seen someone harassed online in some way and 40% have personally experienced harassment themselves. The Centers for Disease Control and Prevention (USA) has reported an estimated 14.9% of high school students have been subjected to cyberbullying in the past 12 months (Committee, 2019). In their study of social media, Oksanen et al. (2014) have reported 67% of 15 to 18 years olds have witnessed cyberbully on Facebook or YouTube, with 21% falling victim themselves.

The large prevalence of toxic comments harms the user experience and also breeds radicalisation, incites violence and triggers real-world tragedy. Evidence has shown that toxic comments on social media such as hate speech have contributed to hate crimes (MacAvaney et al., 2019). For instance, several perpetrators of recent hate-related terror attacks were found to have an extensive social media history of hateful posts, suggesting social media can play a role in radicalisation (Robertson et al., 2018; The New York Times, 2019).

Many countries have already established various relevant laws and regulations to combat violent or bullying language. In England and Wales, relevant legislations include: the Offences Against the Person Act 1861, the Public Order Act 1986, the Malicious Communications Act 1988, the Protection from Harassment Act 1997, and the Criminal Justice Act 2003 (Burnap and Williams, 2016). Germany has enacted the Network Enforcement Act which aims to fight online hate speech (Hilgendorf, 2021). In France, Denmark and the Netherlands, similar laws and regulations are also in place. In the US, there are protections against posting harassing language on the web without exposing personal identity (Burnap and Williams, 2016). In spite of the existence of relevant legislation, toxic comments on social media are still growing and becoming increasingly difficult to control. To combat the plague of toxic comments, platform operators including Twitter, Face-

book, Reddit, and Riot Games (gaming company) have tried to discourage toxic comments by amending their user agreements (Saleem et al., 2017; MacAvaney et al., 2019).

Despite these efforts, toxic comments continue to appear with increasing frequency online, largely due to the perceived anonymity and the difficulty of identifying individuals at scale (Burnap and Williams, 2016). The overwhelming volume of continuously growing UGC online has rendered manual detection of toxic comments infeasible. For this reason, automated identification of toxic comments is becoming an active area of research in both natural language processing (NLP) and machine learning. This is commonly known as toxic comment classification (TCC).

Current methods primarily utilise supervised machine learning (a.k.a. supervised learning) to train a model to identify and classify toxic content. Supervised learning requires labelled data to train the classification model (Schmidt and Wiegand, 2017). That is to say, the classification model needs labelled data (i.e., training data) to learn how to distinguish between toxic comments and normal comments. The training data needs to be labelled manually by humans and this process is laborious and consumes a lot of time and effort. Another key challenge in constructing toxic comments datasets is that what constitutes toxic comments and their categories can be quite subjective, with many competing definitions across legal, regional, platform, and personal contexts (Waseem, 2016a; Rahman et al., 2021). To cope with this challenge, previous studies have trained annotators with specific standards and increased annotator numbers for each example (Waseem, 2016a; Kapil and Ekbal, 2020). However, both approaches increase the cost of creating training datasets. In addition, these labelled training datasets are often ad-hoc and not directly transferable between different tasks or scenarios. At the same, many studies including our previous work have observed that inadequate training data directly leads to the poor performance of classifiers (Deng et al., 2014; Weiss et al., 2016; Pavlopoulos et al., 2017; Zhao et al., 2019; Tan et al., 2018). In short, due to the high cost, it is non-sustainable to create new labelled training data to improve model performance.

Further, training data of TCC tasks are frequently labelled by different schemes (Waseem et al., 2018). For example, they have been labelled using different numbers of labels and different classification rules such as binary classification, multi-

class classification[1] or multi-label classification[2]. The labels are often different too. For example, many papers have previously employed a binary classification with the labels "hate" and "non-hate" or in a similar vein using a kind of toxic comment "X" and its counter "non-X". Examples of these papers are by Burnap et al. (2015), Djuric et al. (2015), Nobata et al. (2016) and Unsvåg and Gambäck (2018). In contrast, in Malmasi and Zampieri (2018)'s research, hate speech is separated from offensive language. They used three labels, "hate", "offensive" and "okay", in a multi-classification task. Other similar multi-classification studies include Davidson et al. (2017) and Badjatiya et al. (2017) and our previous study that categorised text comments into six labels (Zhao et al., 2019).

All these problems mentioned above imply that, despite a significant amount of work on developing TCC training data for different languages, granularity, and tasks, it is often difficult to directly reuse one dataset in another scenario or task in a "like-for-like" fashion (Leon et al.). However, a promising area of machine learning research that looks at how to utilise dissimilar training data or cross-domain model adaptation is transfer learning. Transfer learning transfers knowledge learned by one task to a different task (Lu et al., 2015). The general motivation behind is to utilise more data from different sources to generally improve the target task's performance.

In the last few years, using pre-trained language models is a popular approach to adopt transfer learning in NLP tasks (Ruder12 et al., 2017; Howard and Ruder, 2018; Tan et al., 2018). The framework includes two main parts. First, a language model is pre-trained with unsupervised learning that does not require labelled data. This allows the pre-training to access a huge amount of training data as it does not require human-labelled data. Second, the pre-trained language model is transferred to a downstream task, such as toxic comments detection. The downstream task then can utilise the knowledge learned from the pre-training and tailor the model to its own objective in the second part. This approach has become very popular and in many contexts, replaced the earlier methods of transfer learning (Tan et al., 2018; Devlin et al., 2019; Ruder, 2019). A number of pre-trained language models have been developed and have been made available, such as BERT, RoBERTa, XLM and GPT-2 for general usage, SciBERT, BioBERT and ClinicalBERT for different domains (Devlin et al., 2018; Lample and Conneau, 2019; Liu et al., 2019c; Radford et al., 2019; Beltagy et al., 2019; Alsentzer et al., 2019; Lee

[1]In multi-class classification, there are more than two classes and each instance will be assigned into only one class.

[2]In multi-label classification, there are more than two classes and each instance will be assigned into one or more than classes.

et al., 2020).

However, by the time this work was conducted, only a few studies have addressed TCC tasks with pre-trained language models (MacAvaney et al., 2019). Furthermore, these studies only investigate the performance of BERT with default settings or with only one or two TCC tasks, which leads to limited insights into using pre-trained language models on classifying toxic comments. This thesis is therefore set in this context, and aims to develop understanding, and methods of using pre-trained language models in TCC.

1.2 Research objectives

We aim to investigate current SOTA pre-trained language models (PLMs) on TCC tasks and develop novel methods upon SOTA PLMs to improve the performance on detecting toxic comment.
Our central research question is:

How to effectively use pre-trained language models on TCC tasks?

We will answer this question via addressing three specific questions:

1. How to configure a PLM-based TCC classifier that achieves optimal performance and efficiency?
2. What are the limitations of SOTA PLM-based classifier for TCC?
3. In answering the second research question, we identify that SOTA TCC classifiers tend to suffer from the "identity term bias", in the way that the model tends to predict false positives due to the presence of certain words describing demographic information of individuals or groups. Therefore, our last question is: how to handle the identity term bias found in TCC models?

1.2.1 Configuring PLM-based TCC models

As mentioned in the last section, it is expensive to create more labelled training data to improve TCC model performance. Adopting a pre-trained language model is a method to utilise more data but avoid labelling more data. However, what exact features the pre-trained language model has learned and what features will be transferred to the downstream task are unknown. Therefore, how to adopt a

pre-trained language model and tailoring it to the downstream TCC task is important. While we can always pre-train a language model from scratch, the process will be extremely computationally expensive, requiring a significant amount of resources that may be unavailable to many researchers and practitioners.

For adopting a pre-trained language model, there are several decisions to make in terms of configurations and hyperparameters. How to set up these configurations and hyperparameters is important as this will impact on how the PLM will be tailored to the target task. Also, this directly impacts on the complexity of the final model and therefore, the requirements for computation. For this reason, a study on how these different transferring strategies perform on TCC tasks is needed. Our first research question therefore, particularly looks into this problem.

1.2.2 The limitations of SOTA PLMs

Although deep neural network(DNN)-based models show strong performances on a wide range of NLP tasks, a well-known problem of these models is its lack of transparency in terms of decision making. Previous research has shown that DNN-based TCC models demonstrate bias towards features that should not be linked to toxic comments (Hovy and Spruit, 2016; Blodgett and O'Connor, 2017; Tatman, 2017; Dixon et al., 2018). That is, when a comment contains certain features, such as using African-American English or containing certain keywords, DNN-based models tend to classify it as toxic regardless of its content. This also applies to PLMs (Tatman, 2017; Tan and Celis, 2019; Dixon et al., 2018; Kennedy et al., 2020; Calabrese et al., 2021). A key motivation for developing TCC models is to facilitate content filter and moderation and ultimately foster a friendly online environment between different groups of people (Gallacher, 2021). However, a biased model can potentially cause the conflicts to accelerate. Therefore, this thesis investigates the bias in PLM-based TCC models. We aim to examine the prevalence and the degree of the bias, particularly focusing on identity terms. This is addressed in the second research question.

1.2.3 Addressing the identity term bias in PLM TCC models

Several existing research has studied methods to mitigate the identity term bias in TCC models, which can be categorised into two groups: debiasing the dataset and debiasing the model (Dixon et al., 2018; Davidson et al., 2019; Wiegand et al.,

2019). Methods of debiasing the dataset aim to curate the dataset via different approaches to "remove" bias-related features. However, it has been argued that data curation may introduce new bias into the dataset and there is no standard on the degree to which a dataset can be modified (Clark et al., 2019; Rogers, 2021; Ribeiro et al., 2020; Garrido-Muñoz et al., 2021). On the other hand, methods of debiasing the model aim to modify the model to mitigate the identity term bias. These methods mainly follow the same principle, to suppress the model's attention to identity terms. However, identity terms are sometimes important for classification decisions and therefore, ignoring them is not an optimal solution. This thesis aims to develop novel methods that mitigate the identity term bias and this is addressed in the third research question.

1.3 Ethical considerations

While we believe this thesis will provide important insights on the development pathways of combating toxic comments online, it may raise concerns over ethical implications, such as potentially being able to breach the privacy of toxic comment authors. We would like to emphasise that all data used in this thesis are secondary data published by previous studies. Also, we did not attempt to use any individual-level information in our study. Yet, we aim to predict and detect toxic comments, but only analyse the textual contents of comments. The Research Ethics Review documentation and approval are attached in the appendix.

1.4 Thesis structure

Chapter 2 gives an overview of background information, concepts and techniques that are relevant in order to understand the contents of this thesis. We review literature pertaining to TCC with a focus on classification approaches. Here, we will introduce essential concepts and techniques in text classification, machine learning and NLP that are required for this research.

Chapter 3 investigates different strategies of transferring PLMs to TCC models. Especially, we explore the design of downstream architectures and the continued pre-training in domain. For both angles, we study the importance of transferring efficiency. That is, maintaining comparable performance while minimising the requirements for computational resources.

Chapter 4 examines the identity term bias in PLM-based TCC models on a wide range of TCC tasks. This chapter also seeks to provide further evidences

for the identity term bias with model explanations. Further, we conduct a novel analysis of the identity term bias which links the bias to the subjectivity level of comments.

Chapter 5 proposes a novel method to handle the identity term bias in PLM-based TCC models. The method is evaluated on a wide range of TCC tasks and compared to a range of SOTA models to provide deep insights of the identity term bias issue.

Chapter 6 concludes this thesis with a summary of our main contributions, discusses limitations of this research, and identifies future research directions.

1.5 Publications

The studies in this thesis mainly reflect the following peer-reviewed articles (in order of publication date):

Zhao, Z., Zhang, Z., Hopfgartner, F. (2019). Detecting Toxic Content Online and the Effect of Training Data on Classification Performance. In Proceedings of 20th International Conference on Computational Linguistics and Intelligent Text Processing (Chapter 3)

Zhao, Z., Zhang, Z., Hopfgartner, F. (2021, April). A comparative study of using pre-trained language models for toxic comment classification. In Companion Proceedings of the Web Conference 2021 (pp. 500-507). (Chapter 3)

Zhao, Z., Zhang, Z., Hopfgartner, F. (2022). Utilizing subjectivity level to mitigate identity term bias in toxic comments classification. Online Social Networks and Media, 29, 100205. (Chapters 4 and 5)

While not directly related, the following articles have come across to this thesis in terms of methodology. They have been published over the course of the PhD:

Clowes, M., Stansfield, C., Thomas, J., Shemilt, I., Paisley, S., Stevenson, M., **Zhao, Z.**, Marshall, I., Kell, G., (2022). All is FAIR in health inequalities research: using machine learning to build a new database of health equity studies. European Association for Health Information and Libraries 2022. (Section 5.3.3)

Zhao, Z., G. Chrysostomou, K. Bontcheva and N. Aletras (2022). On the

Impact of Temporal Concept Drift on Model Explanations. In Findings of the Association for Computational Linguistics: EMNLP 2022.(Section 4.4.2)

Chapter 2

Literature review

TCC is a NLP task that aims to detect toxic comments based on their text content. There are several other tasks and research areas related to TCC, such as sentiment analysis that detects the implied sentiment of a text (Dos Santos and Gatti, 2014; Touahri and Mazroui, 2021); sarcasm detection that identifies sarcastic behaviour in the field of online social networks (Amir et al., 2016; Felbo et al., 2017; Javdan et al., 2020; Ashwitha et al., 2021) and misinformation detection that identifies, for example, fake news, click-bait, and false rumours (Jiang and Wilson, 2018; Lee et al., 2021; Schroeder et al., 2021). In addition to investigating toxic comments on the content level, some research focus on toxic user detection and toxic comment diffusion analysis (Ribeiro et al., 2017; Mall et al., 2020; Gallacher and Bright, 2021). Tasks discussed above and TCC tasks have some shared concepts, methods and techniques. This chapter does not cover these areas in detail, but only focuses on content level TCC.

This chapter gives an overview of TCC tasks, covering its definition, methodology and challenges, to set the stage for the subsequent chapters. The purpose of this chapter is to prepare the necessary knowledge and essential background for this thesis, while in later chapters, we will cover more specific areas of research that are more closely related to the topics of those chapters. We start by firstly introducing the text classification framework in Section 2.1. This is because TCC tasks are text classification tasks essentially, and they follow the same framework of text classification methods. Then we review two fundamental concepts in text classification: features and algorithms. Commonly-used features for TCC are introduced in Section 2.2. Features are used by algorithms to represent data, and can be grouped into manually engineered features and abstract features. Commonly-used algorithms for TCC are divided into two groups that are discussed seperately in Section 2.3. The first group is traditional machine learning algorithms, which

conventionally work in concert with manually engineered features; the second one, neural network-based algorithms, learns abstract features automatically during the training process (Zhang et al., 2018c). Section 2.4 focuses on transfer learning, which is a SOTA NLP framework. Transfer learning forms the cornerstones of the techniques studied throughout this thesis. Section 2.5 introduces the concept of pre-trained language models (PLMs), which are nowadays the mainstream approach to transfer learning. The last part of this chapter, Section 2.6, summarises research gaps in TCC that are relevant to this thesis.

2.1 Task definition

2.1.1 Text classification overview

Text classification is a common task in NLP. It aims to classify given texts or documents into appropriate categories according to their content (Uysal, 2016). One typical classification task is binary classification, where one data instance (i.e., a text or a document) is classified as one or the other, e.g., yes or no, positive or negative, and toxic or not toxic. In addition to binary classification, multi-class classification and multi-label classification are another two common classification frameworks. Multi-class classification tasks assign each data instance into one out of multiple (more than two) classes. In multi-label classification, one instance can be associated with one or more classes.

The appropriate classes are usually pre-defined by human annotators, which makes text classification a supervised learning task. In supervised learning, training data are labelled by humans and therefore, also called "labelled data" and "gold standard". These are used to train a machine learning model. Unsupervised learning, on the other hand, does not require labelled data. Intuitively speaking, the learning process of the classification model is "supervised" by the training data which has "golden standard" labels, i.e., the labels assigned by human to each text or document. The goal of the learning/training is to make a prediction of labels as close to the "golden standard" as possible. A conventional framework of supervised text classification task consists of **data pre-processing**, **feature extraction and representation**, **model training** and **model evaluation** (Schmidt and Wiegand, 2017).

The purpose of **data pre-processing** is to reduce the noise and complexity in the text to make it easier for classifiers to decipher and process the text (Uysal and Gunal, 2014). Common pre-processing techniques are stop-words removal,

Actual label \Predicted label	Positive	Negative
Positive	True positive (TP)	False negative (FN)
Negative	False positive (FP)	True negative (TN)

Table 2.1: *The confusion matrix of classification prediction*

stemming and lemmatization (Manning et al., 2010). Stop-words are frequently-appearing and meaningless words that have low discriminative power, such as "the", "a" and "that" (Lo et al., 2005). Stemming and lemmatization both transform words in inflectional forms or derivational forms to a root form (Manning et al., 2010). For examples, *unhappy* and *happiness* are derivational forms of the root word *happy*. Therefore, if applying these techniques, the sentence "They're just beasts walking on two legs" could be "beast walk two leg" (Manning et al., 2010). "They're", "just" and "on" have been removed as stop-words, "beasts", "walking" and "legs" have been transformed to their root form by stemming and lemmatization.

Feature extraction and representation aims to represent textual data with features, usually using numerical vectors or matrices (John, 2017). In machine learning, a feature is an individual measurable property of an instance which is used to differentiate instances. A range of features constitutes a feature set, deciding how an instance is interpreted by the classifier. Selecting discriminative and informative features is crucial for training a model to effectively capture regularities of different categories. In the context of text classification, features can be words, phrases, part-of-speech and so forth. These features are further explored in Section 2.2.

The next step is **model training** when a defined algorithm is trained on a training dataset with golden standard labels. In this step, the model learns to generalize patterns over features observed in the training data. The derived patterns are expected to capture "enough" characteristics of unseen data for classification. Different algorithms for text classification will be given more details in Section 2.3

The final step is **model evaluation** which evaluates the performance of the model with testing data, i.e., unseen data during training. The idea is to compare the predicted labels by the model with the golden standard labels assigned by annotators. Predictions by a model can be categorised into 4 groups as shown in the confusion matrix, (shown in Table 2.1). Common evaluation metrics include recall, precision and F_1 score. Recall and precision are calculated as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.1}$$

As indicated in Equation 2.1, recall is the proportion of real positive cases that are correctly predicted as positive (Powers, 2011; Buckland and Gey, 1994). It measures the coverage of the real positive cases by the model. Precision is the proportion of predicted positive cases that are real positives (Powers, 2011; Buckland and Gey, 1994). F_1 score considers both recall and precision. The traditional F_1 score or balanced F_1 score is the harmonic mean of precision and recall, calculated as:

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})} \quad (2.2)$$

In the context of multi-class and multi-label classification, the evaluation metrics need to take different labels into account. That is, the prediction of one data instance can be true positive to one class and true negative to another class, false positive to the third class. Different approaches have been designed to handle the evaluation for multi-class and multi-label classification, such as hamming loss, micro-average, macro-average and ranking loss (Zhang and Zhou, 2013; Sorower, 2010; Prajapati et al., 2012). We introduce micro-average and macro-average as they are widely-used in text classification tasks (van Aken et al., 2018; Qian et al., 2018; Wiegand et al., 2019; Ousidhoum et al., 2019; Zhao et al., 2019; Fagni et al., 2019; Agarwal and Chowdary, 2021; Zhao et al., 2021). A macro-average computes the metric independently for each class and then takes the average, whereas a micro-average aggregates all categories' contributions to computing the average metric (Agarwal and Chowdary, 2021; Zhang and Zhou, 2013; Sorower, 2010). Therefore, macro-average treats all categories with equal importance. The ability of a model to behave well on categories with few positive examples, i.e., minority categories, will be emphasised by macro-average metrics and much less so by micro-average metrics.

2.1.2 Toxic comment classification

The detection of toxic comments can be treated as a text classification problem, which intends to classify comments into different categories, which are categories of toxic comments (e.g., "sexist", "racist", "hate speech", and "abusive language") and categories of non-toxic comments (i.e., "clean" or "healthy" comments), following the framework of text classification discussed above (Schmidt and Wiegand, 2017; Zhang et al., 2018c; Zhao et al., 2019). There is a wealth of research on detecting different kinds of toxic comments such as hate speech, cyberbullying and

abusive language.

The majority of research focuses on building a binary classifier to separate one particular type of toxic comment from all the other comments (Fortuna and Nunes, 2018). Kwok and Wang (2013), Burnap et al. (2015) and Djuric et al. (2015) apply binary classification methods to classify each text into groups of either "hate speech" or "non-hate speech". Similarly, Nobata et al. (2016) and Che et al. (2017) formulate binary classification tasks to distinguish between abusive language and clean language, Dinakar et al. (2011) develops a binary classifier to detect cyberbullying comments.

A few papers treat TCC as a multi-class or multi-label classification problem. For example, Park and Fung (2017) and Waseem and Hovy (2016) categorise comments into three groups, "none", "racism" or "sexism", where each comment belongs to only one class. In multi-label classification, one instance can be associated with one or more labels. For example, in our earlier study, there are six labels together, namely "severe toxic", "toxic", "obscene", "threat", "insult" and "identity hate" (Zhao et al., 2019). Each data instance can be assigned with zero to six of those labels at the same time. Therefore, a comment could be "toxic", "insult" and "identity hate" or other combinations of labels. However, if treated as a multi-class classification problem, it could only be either "toxic", "insult" or "identity hate".

Beyond challenges faced by generic text classification tasks, such as ambiguous words and underrepresented classes, TCC faces additional difficulties due to its nature (Wang et al., 2014; Taghipour and Ng, 2015; Nobata et al., 2016; Collins et al., 2018). For one, toxic comments stem from user-generated contents, which usually contain a lot of informal usage of languages, such as slang, cyberspeak, typos, variants and emoji (Nobata et al., 2016). For instance, some posts replace letters with similar looking numbers ("E"s with 3s, or "l"s with 1s, and so on). Variants like this are extensive and include various combinations. Another issue is that the toxicity of a comment largely depends on its context. Even many expressions that are not inherently toxic can be toxic in some context, or vice versa (ElSherief et al., 2018; Ullmann and Tomalin, 2020; Kovács et al., 2021). For example, the usage of "nigger" within the group of Africa-American (also known as in-group usage), e.g., an Africa-American uses it to refer to another Africa-American, is usually not considered hateful; however, out-group usage, e.g., a white American uses it to refer to an Africa-American, is deemed hateful (Halevy et al., 2021). What is more, TCC usually handles short text, which provide less information for the model to learn. Further, different TCC tasks suffer from a

similar issue that the target behaviour, such as hate speech or cyberbullying, can be subtle and indirect (Alorainy et al., 2019). Therefore simple methods that use simple lexical features (e.g., words or phrases), can lead to a significant number of false negatives (Alorainy et al., 2019).

2.2 Features

Before discussing different features, we first briefly introduce how to encode features. One-hot encoding is a basic approach which uses binary identifiers (0 or 1) to indicate the presence or absence of a feature where each feature is represented as a unique dimension (Goldberg, 2016). To illustrate, we set up a micro pseudo dataset which only contains four sentences as shown in Figure 2.1. In this dataset, the corpus has 15 unique words and each one is represented as a single feature, i.e., a dimension. The dimension of of feature vectors is equal to the size of the vocabulary of the corpus, 15. As shown in the figure, the representation for "good to know" has three "1"s for the words it contains, coupled with twelve "0"s indicating the absence of the remaining words in the corpus.

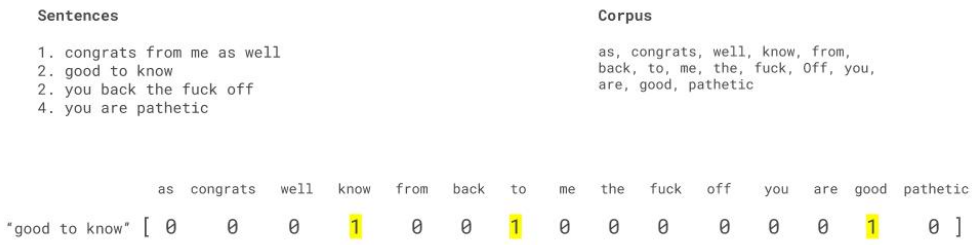

Figure 2.1: *An example of one-hot encoding: representing the sentence of "good to know" in a pseudo dataset which only contains 4 sentences and 15 words.*

In practice, the training process involves much larger datasets than the one in Figure 2.1, consisting of thousands or even millions of instances. This creates a very large feature space of high dimensionality. Each instance will have a large dimensional vector with only a handful of non-zero values, resulting in a very sparse feature space, known as sparse representation. The storage and computation of sparse representation are extremely ineffective (Mikolov et al., 2013).

To address this issue, distributed representations of words have been developed to represent words and texts with dense real-valued tensor (e.g., vector and matrix) (Bengio et al., 2003; Mikolov et al., 2009; Pennington et al., 2014; Devlin et al.,

2019). It is more often referred to as word embeddings (Bengio et al., 2003; Mikolov et al., 2009; Pennington et al., 2014; Devlin et al., 2019; Founta et al., 2019). In short, word embeddings represent textual instances with low-dimensionality tensors and each element of the tensors is a non-zero value. These values are difficult to interpret intuitively. In other words, these values represent abstract features of a given textual instance. More details regarding word embeddings are discussed in Section 2.2.2 as it is a key concept to this thesis.

The rest of this section introduces different features used to represent textual instances, which have been grouped into manually engineered features and word embeddings (distributed word representations) (Tang et al., 2014; Zhang et al., 2015). Manually engineered features are concrete inputs which are easy to interpret, in contrast to the demanding interpretability of word embeddings (Goldberg, 2016).

2.2.1 Manually engineered features

Surface features

N-gram is a single item or a continuous sequence of items from a given word or text document. In other words, the n-gram model divides a word or a text into n-chunks of items. For example, character level bigrams (2-grams) for *"hate"* are *ha*, *at*, *te*; word level bigrams for *"I hate you"* are *"I hate"*, *"hate you"*.

Skip-gram is a variation of n-gram, which divides a word or comment by skipping one or several items, capturing the information from non-contiguous and long distance items (Chavan and Shylaja, 2015). To be more specific, a skip-gram is defined in the format of k-skip-n-gram, referring to skip k and also less than k items and construct the n-gram (Guthrie et al., 2006). As such, "2-skip-n-gram" results include 2 skips, 1 skip, and 0 skips. For example, 2-skip-2-grams for *"fuck off u gay boy"* include 2 skips = *fuck gay, off boy*, 1 skip = *fuck u, off gay, u boy*, 0 skip = *fuck off, off u, u gay, gay boy*. It is widely agreed that by jumping over words, skip-grams capture the longer distance dependencies between words (Guthrie et al., 2006; Chavan and Shylaja, 2015; Malmasi and Zampieri, 2018).

Term Frequency-Inverse Document Frequency (TF-IDF) evaluates the importance of words or items in a text document based on how frequently they appear across the whole training dataset (Sparck Jones, 1972; Rajaraman and Ullman, 2011). TF measures the frequency of a given word in a document. TF of a word will be different depending on which document it is in. IDF assesses the importance of the words by giving an inverse proportional score to the number

of documents in which a word is found, indicating how frequent a word appears in the entire training dataset. Therefore, the classifier will focus on more informative words. While words with less discriminative power, such as "the", "a", "is", although appearing often, will be given low IDF scores and thus be given low TF-IDF scores. In short, TF-IDF of a word reflects the informativeness and importance of a word and is decided by its frequency in the document (the data instance) and the whole dataset.

N-grams and TF-IDF are usually applied together. To be more specific, each n-gram is presented to the classifier with a TF-IDF weight. In this way, certain expressions with high TF-IDF tend to give a greater impact on the classification prediction. The combination of N-grams and TF-IDF is often reported to be very effective and thus has been included in the feature sets of a wide range of research (Schmidt and Wiegand, 2017). For example, in the task of hate speech classification, Waseem (2016b) and Saleem et al. (2017) both incorporate uni-gram and TF-IDF weights. In addition to uni-gram with TF-IDF, Chavan and Shylaja (2015) also adds skip-gram with TF-IDF to enhance their feature set.

Sentiment analysis features

It is often safe to assume that hate speech carries negative connotations and therefore, negative sentiment (Schmidt and Wiegand, 2017). Consequently, toxic content detection and sentiment analysis are closely related. Researchers commonly utilize sentiment analysis to derive latent sentiment features for TCC. Typically, sentiment analysis leverages the type of polarity (i.e., positive and negative; subjective and objective) expressed in a message as well as polarity intensity (Gitari et al., 2015; Schmidt and Wiegand, 2017).

In the task of detecting cyberbullying, Van Hee et al. (2015) incorporate sentiment features by counting the number of positive, negative and neutral words in a given text. Gitari et al. (2015) suggest that hate speech is laden with heavily subjective expressions. Thereby, the presence of objective or subjective expressions in a text is taken as a feature in their study of hate speech classification.

Lexical resources

Most toxic comments contain specific vocabulary such as slurs and insults (Schmidt and Wiegand, 2017). For this reason, the presence of such words can be a predictive feature for detection. To capture these words, a task-specified vocabulary

is required. There are several publicly available lexical resources[1] related to toxic comments which could be predictive (Schmidt and Wiegand, 2017). For instance, Hatebase is a lexicon focusing on hate-related terms. It includes 3879 terms regarding hate speech towards different nationality, ethnicity, religion, gender, sexual discrimination, disability and minority groups (Kennedy et al., 2018). HurtLex is a hand-crafted lexicon of offensive, aggressive and hateful words. Its original version is in Italian and then it is semi-automatically translated into 67 languages (Bassignana et al., 2018). Off-the-shelf lexical resources are convenient and have been employed by several papers of toxic comment detection (Xiang et al., 2012; Burnap et al., 2015; Pamungkas et al., 2021).

Some researchers have compiled word lists as their lexical resources for specific tasks rather than utilising the off-the-shelf lexical resources (Razavi et al., 2010; Gitari et al., 2015; Alorainy et al., 2019). For example, for building a classifier to detect hate speech, Gitari et al. (2015) develop a lexicon by compiling "hate"-related verbs and incorporated them as additional signals in features. This is because some verbs are commonly used to condone and encourage violent acts in hate speech, such as "*loot*", "*riot*", "*beat*", "*kill*", and "*evict*" (Gitari et al., 2015). This view and observation is supported by the research of Wester et al. (2016) that investigates the effect of lexical, syntactic and semantic features on detecting threats of violence messages. According to their results, the combinations of lexical features outperform the use of more complex syntactic and semantic features, highlighting the effectiveness of lexical resources in detecting violence-related toxic comments (Wester et al., 2016).

Linguistic features

Due to the complexities in language grammar and syntactic relationships, it is common for the same words to have different meanings when they appear in different orders. Therefore, linguistic knowledge can be introduced to indicate word relationships. Part-of-speech (POS) tagging and typed dependency parser are two popular linguistic features.

POS tagging is the process of marking up a word in a text as corresponding to a particular word category that indicates its syntactic role, such as plural noun, adverb and adjective (Behzadi, 2015). POS tagging has been used for a variety of NLP tasks to provide the linguistic signal. In the context of toxic content classification, researchers often add POS tagging to enrich the feature set of their

[1]HurtLex: http://hatespeech.di.unito.it/ (accessed on 1 November 2021); Hatebase: https://hatebase.org/ (accessed on 10 November 2019)

model (Dinakar et al., 2011; Nobata et al., 2016).

Typed dependency relationship analyses the grammatical structure of a sentence, identifying relationships between words. For instance, this encompasses groups of words which go together as "phrase" and differentiates words as the subject or object of a verb. Typed dependency can also preserve word order in sentences and uses it as a feature for models (De Marneffe and Manning, 2008). Overall, typed dependency relationship aims to extract textual relations and use them as features in the task. For example, in the tweet, "*Totally fed up with the way this country has turned into a heaven for terrorists, Send them all back Home*", one resultant typed dependency would be the nominal subject between the fifth word "*home*" and the second word "*them*" (Burnap and Williams, 2016; Fortuna and Nunes, 2018).

Othering language refers to the use of language to express divisive opinions between the in-group (e.g., "us") and the out-group (e.g., "them"). Othering is an established construct in rhetorical narrative surrounding hate speech, and the 'we-they' dichotomy has long been identified in racist discourse (Meddaugh and Kay, 2009; Wodak and Reisigl, 1999; Burnap et al., 2015). A few previous work have used othering language as an feature for TCC tasks (Burnap et al., 2015; Burnap and Williams, 2016; Alorainy et al., 2019). In their study, Alorainy et al. (2019) identify the othering feature by the use of two-sided pronouns that combine the in-group and out-group (e.g., your/our, you/us, they/we), and the use of pronoun patterns, such as verb-pronoun combinations, which capture the context in which two-sided pronouns are used (e.g., send/them, protect/us). Example toxic comments to meet their hypothesis can be "we want to send them home" and "we do not need them to take our jobs". They have found that adding this feature set improves the model performance on cyberhate detection tasks. Burnap et al. (2015) have also found that othering language is a useful feature for classifying cyberhate based on religious beliefs, specifically for identifying anti-muslim sentiment.

User-based features

Features discussed above are all content-based, which only analyse the comments themselves. As summarised by Salawu et al. (2017), more than 41 papers use content-based features to detect cyberbullying, indicating their crucial role. In contrast, user-based features utilise the user/publisher (i.e., who post the comment) information, such as their age, gender, location, number of friends and

social network, conventionally seen as auxiliary features of content-based features (Nahar et al., 2014; Huang et al., 2014). It is believed by a few papers that the background information of the content author is predictive for the detection of toxic content (Schmidt and Wiegand, 2017). A simple example is that if users are known to previously post toxic messages, they may post them again. Furthermore, in many cases, toxic comments cannot be determined solely based on the text content (Founta et al., 2019). For instance, "*Put on a wig and lipstick and be who you really are*" may not be regarded as a form of toxic comment when read in isolation, however, it can be offensive when the context information is given that this utterance has been directed towards a boy. As one could infer that this is a remark to malign the sexuality or gender identity of the boy being addressed (Schmidt and Wiegand, 2017).

Waseem and Hovy (2016) tested the effect of user-based features, gender, location and the word length, in addition to simple surface features in TCC tasks. Their results show that gender is the only additional feature that improves the performance of detecting offensive tweets. They also identify a pattern which shows that the gender distribution of hate speech in their dataset is heavily skewed towards men. This may explain the accuracy improvement gained with the addition of gender information. Similarly, by adding the gender feature, Dadvar et al. (2012) observes improvements in their classification model of detecting cyberbullying disclosures.

Also in the field of cyberbullying detection, Chatzakou et al. (2017) consider network-based user information measuring the power imbalance between bully and victim, as well as the bully's popularity based on interaction graphs and the bully's position in the network.

2.2.2 Abstract features - word embeddings

Feature representation techniques above take individual features as discrete items (Mikolov et al., 2013). In other words, these representations are arbitrary, ignoring the semantic relationships and latent relationships between features. For example, when adopting one-hot encoding as described above, "*Muslim*" and "*Paki*" are represented as two independent features, which does not indicate the latent relationship between them as shown in Figure 2.2. This means that the model will treat "*Muslim*" and "*Paki*" separately, and what the model learned about "*Muslim*" would not be leveraged when processing "*Paki*". Another similar example could be "*African*" and "*Nigger*".

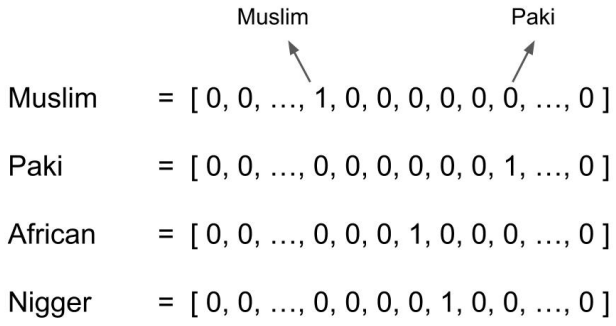

Figure 2.2: *An example of feature representations: Representing individual words in vectors with one-hot encoding, where only unigram is considered.*

Another limitation of these feature representations is their high dimensionality. High-dimensional representations have historically presented serious challenges to machine learning models and the learning process. This is often referred to as the *curse of dimensionality* (Aggarwal, 2014). Back to the example in Figure 2.1 on page 15, if the corpus has 20,000 unique words and phrases, then "good to know" will be represented by a 20,000 dimensional vector with only three "1" and 19,997 "0", which is extremely ineffective to compute and process. Furthermore, these dominating '0's may have an effect of "diluting" the really discriminative features (Aggarwal and Zhai, 2012a). In short, a high number of dimensions leads to complex computation and makes the prediction more difficult.

Several distributed word representation methods, also known as word embeddings, have been developed to capture the relationships between words or phrases but also tremendously decrease the number of dimensions. A distributed representation of a word is a vector (usually hundreds of dimensions) of continuous real numbers which characterise the meaning of the word. Thus, a sentence or a text of any length can be presented by a combination of these feature vectors. Figure 2.3 shows a pseudo representation of a sentence with 3 dimensional vectors.

Intuitively, one word vector corresponds to a point in a feature space, where each dimension of that space corresponds to semantic or grammatical characteristics of that word (Deng, 2014). As shown in Figure 2.4, similar or related words are close to each other and certain semantic relationships between words pairs are shown by following similar directions (Deng, 2014). The ideal hypothesis behind this is that the geometric relationships in the feature space between two vectors tend to indicate their semantic relationships.

In this way, the distributed word representations of words with related seman-

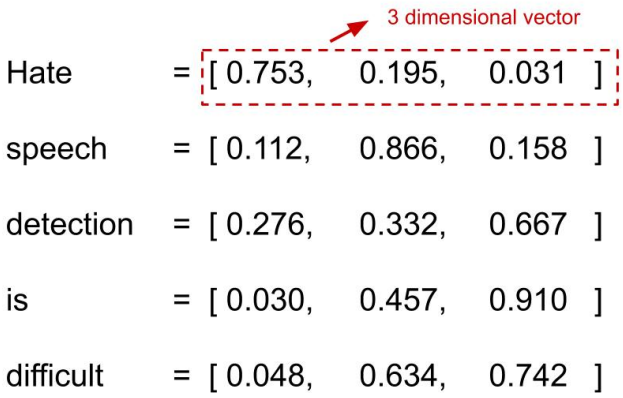

Figure 2.3: *A representation of a sentence with pseudo 3-dimensional distributed word representations, where each word is represented by a 3-dimensional distributed vector and the sentence is therefore a 3*5 matrix.*

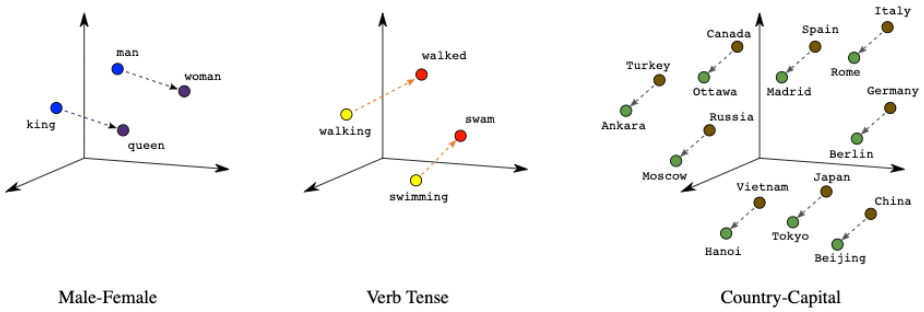

Figure 2.4: *Visualizations of distributed word representations in vector spaces, i.e., word vectors (Google, 2019)*

tic meanings tend to be similar. Having said that, what each dimension exactly means is difficult to interpret. But every dimension could be considered as a "latent" concept or a combination of several topics, and hence "abstract" features (Mikolov et al., 2013). By contrast, manual feature engineering combines all selected features into the input vectors or matrices and then directly feed them into the classifier, preserving a high-level interpretability (van Aken et al., 2018).

Context-independent embeddings

Word embeddings can be obtained via pre-training on a large corpus and then used directly to represent words in downstream tasks. In this way, word embeddings are stored in a look-up table and are independent from the context. Word2Vec and GloVe are two prominent context-independent word embedding models (Mikolov et al., 2013; Pennington et al., 2014; Camacho-Collados and Pilehvar, 2018). Their

distributed word representations are trained on statistical language models, which are based on the co-occurrence of words, phases or symbol combinations and then mapped the derive high-dimension vectors into low-dimensional vectors by different methods. Their approach is in accordance with the distributional hypothesis proposed by Harris (1954), that words are similar if they occur in similar context and the semantic meaning of the word can be indicated by its context.

Word2Vec is trained on language modelling tasks which aim to predict the centre word based on the context (Continuous Bag-of-Words Model, as denoted as "CBOW") or to predict the context of the centre word (Continuous Skip-gram Model), as shown in Figure 2.5. From the perspective of model training, the objective function of Word2Vec is to maximise the log probability of context words W_c given its input word W_t, i.e., $\log P(\mathrm{W}_c|\mathrm{W}_t), where \mathrm{W}_c$ refers to W_{t-2}, W_{t-1}, W_{t+1}, W_{t+2} or any combination of them. Each word is mapped to a unique vector, represented by a column in the matrix of the text. The word vectors are initialised with random values and then those values are adjusted to optimise the prediction during the training. By training, those word vectors eventually capture semantics and the final word vectors are the by-product of the prediction task (Le and Mikolov, 2014).

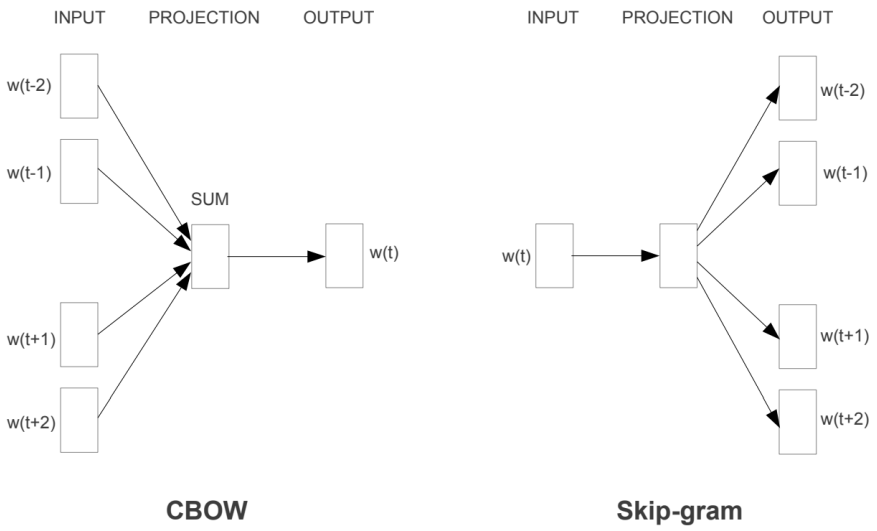

Figure 2.5: *Two different versions of Word2Vec proposed by Mikolov et al. (2013). W_t is the center word. W_{t-1} and W_{t+1} is the word before and after centre word.*

In contrast to the "predictive" model of Word2Vec, **GloVe** is a count-based statistic language model, counting the co-occurrence of words in the context (Pennington et al., 2014). It learns to construct aggregated global word-word co-

occurrence matrix (word-word) to reflect how frequently a word appears in a context (Pennington et al., 2014). The distributed word vector could be taken as a by-product of the construction of the co-occurrence matrix.

Word2Vec and GloVe are the early, seminal work for context-independent embeddings. There are later works (e.g., Bojanowski et al. (2017) proposes FastText) that we do not cover in details, because the fundamental ideas behind them are similar and also the current trend shifts to context-sensitive embeddings which will be covered in the next section. A limitation of context-independent embeddings is that it treats polysemous words (words with multiple meanings) as a single entity. In other words, context-independent embeddings assign each word a static representation regardless of the context and the word meaning in the sentence, but polysemous words have different meanings in different contexts (Deng, 2014). In this way, the word embedding for "Nice" is the same regardless if its meaning ("good" or the French city). In this way, the different senses of a word are encoded into one vector.

Context-sensitive embeddings

To dynamically represent words in different contexts, context-sensitive embeddings were proposed so that the word embeddings change according to their contexts. CoVe and Elmo are two popular embedding systems belonging to this branch (Deng, 2014).

CoVe stands for Contextualized Word Vectors, derived from a English-to-German machine translation model (McCann et al., 2017). The sequence-to-sequence machine translation model consists of an encoder and a decoder (Sutskever et al., 2014). The encoder is pre-trained to encode the sequence of words, i.e., a sentence, from the original language (English) to an "intermediate language" (also can be thought as a intermediate representation). The decoder is to decode the "intermediate language" into the target language (German). The pre-trained encoder is then transferred to the downstream model as additional functions for the input embeddings. The output of the encoder is sequences of contextualized vectors. The final CoVe representation of a word is composed by concatenating its corresponding Glove Vector and its contextualized vectors derived from the pre-trained encoder output (McCann et al., 2017).

The intention behind the GloVe-CoVe concatenation is intuitive, so to let the GloVe vector provide the basic semantic information of the word and let the context vector give the information of context. To be more specific, the encoder

processes and encodes the text at a sequence level, the output is more sentence-specific and therefore gives more information about the context. From a technical perspective, contextual vectors in the CoVe word representations are functions of the entire input sentence, rather than fixed vectors like context-independent embeddings.

Elmo is a newer context-sensitive word embedding model proposed by Peters et al. (2018a) one year after CoVe was published. Like CoVe, Elmo is derived from a pre-trained model, and is subsequently transferred to the downstream task. Likewise, both models use layer-wise neural network algorithms which will be explained in Section 2.3. The main difference is that Elmo trains on an unsupervised language model rather than a supervised translation model utilized by CoVe.

In summary, context-independent embeddings, Word2Vec and GloVe, serve as dictionary resources to the model which can also be thought as static lookup tables. On the contrary, CoVe and Elmo are dynamically generated by incorporating contextual text. Rather than having a dictionary 'looking up' corresponding vectors for each word, CoVe and Elmo create vectors on-the-fly by passing input text through their pre-trained neural networks. Alternatively, they are also taken as pre-trained models, also known as a form of transfer learning (Devlin et al., 2018).

Take word representations in Figure 2.3 as an example, when using context-independent embeddings, the word embedding for word *"hate"* is fixed as $[0.753, 0.195, 0.031]$ no matter what words surround *"hate"*. On the contrary, if using context-sensitive embeddings, such as CoVe and Elmo, the representation for *"hate"* will not be $[0.753, 0.195, 0.031]$ but changes dynamically according to its context.

From word embeddings to paragraph embeddings

Since in TCC, sentences or paragraphs are classified rather than words, representation methods for sentences and paragraphs under the auspices of word embedding are sought (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018).

A straightforward method is averaging or summing the word embedding of all words in the sentence or the paragraph and taking the averaged or summed vectors to represent the sentences or paragraphs (Wieting et al., 2015; Arora et al., 2016; Nobata et al., 2016; Arora et al., 2016; Lee and Yoon, 2018; Fortuna and Nunes, 2018). As shown in Figure 2.6, when applying the averaging method (highlighted in grey), the sentence vector dimension is equal to the word vector's dimension. The same situation goes for the summing method.

Theoretically speaking, the averaged/summed representation loses word order

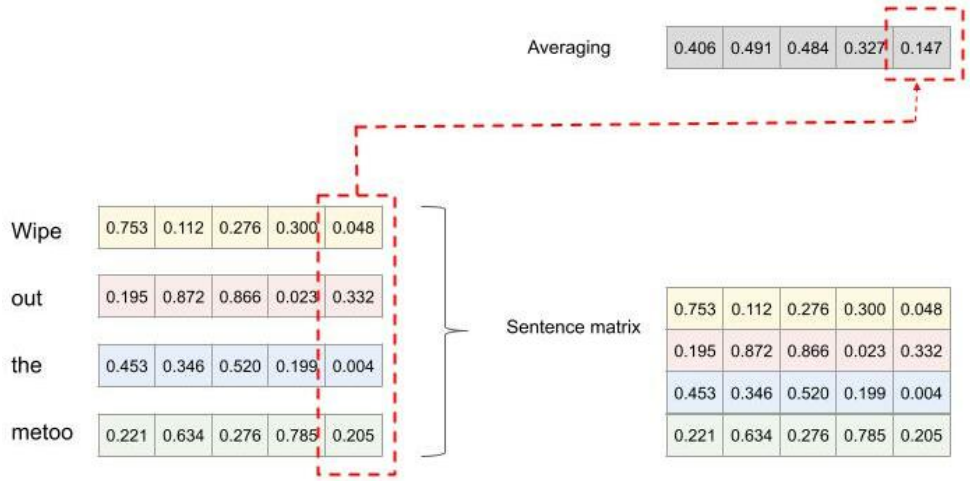

Figure 2.6: *An illustration of composing a sentence vector by averaging word vectors and composing a sentence matrix).*

sensitivity in the same way N-gram models do (Le and Mikolov, 2014). However, a few papers have shown that such a simple method works as good as or better than other feature representation methods. In the task of detecting abusive language, Nobata et al. (2016) have tested a group of features including averaged representation which averages the word embeddings (Word2Vec) of all words in the comment. The average of all word embeddings were reported to be outperformed by word-level N-grams and character-level N-grams (Nobata et al., 2016). Arora et al. (2016) proposed an "embarrassingly simple" sentence embedding method: a weighted sum of word vectors (GloVe). They weighted each word vector by the factor $\frac{a}{a+p(w)}$ where a is a hyper-parameter and $p(w)$ is the (estimated) word frequency. This is similar to IDF weighting, where more frequent words and terms in the corpus are weighted down.

Another simple and widely-used approach is combining word vectors for each word in a sentence into a sentence matrix as shown in Figure 2.6 (Kim, 2014). In this way, each row corresponds to a word in the sentence and the order of rows follows the order of words in each sentence. This approach is also called concatenation. In the spatial 2-dimensional matrix, the horizontal rows capture each word's semantic meanings. Additionally, the vertical order of rows indicates the words order that the averaged vectors and summed vectors discussed above are unable to capture. However, the matrix approach may lose information when using arbitrary sentence lengths since useful contents may be truncated. More specifically, every text fed into the model need to be converted to a matrix of a identical dimension. Therefore, the number of words in each text need to be the

same for a identical vertical size. That is, when a sentence is shorter than the predefined length, it will be padded with zero values, when a sentence is longer than the length, it will be truncated to the length.

The methods above manipulate word embeddings to formulate new embeddings for whole sentences or paragraphs. In addition to these approaches, Le and Mikolov (2014) have proposed *paragraph vector (para2vec)*, which learns continuous distributed vector representations for pieces of texts from scratch. Their technique was inspired by Word2Vec by training the model to predict a certain word based on surrounding words and then extracting word vectors generated in the prediction task as word embeddings. The idea was extended to get paragraph vectors in a similar manner (Le and Mikolov, 2014). They established a prediction task much like Word2Vec but with the addition of a paragraph vector alongside word vectors. After training the prediction model, the paragraph vector is pulled out to represent the paragraph, which can then be intuitively taken as the "byproduct" of the prediction task.

Para2Vec can potentially address limitations of averaging/summing approach and matrix approach, such as losing information of words order and unable to take arbitrary lengths of sentences. Further, Lau and Baldwin (2016) empirically shows that para2vec requires a large external corpora for training for comparable performance. However, Para2vec cannot dynamically compose vectors for unseen sentences or paragraphs in the training data.

2.3 Algorithms

This section introduces algorithms, an essential aspect of text classification in conjunction with features. Similar to the previous section on features, this section also separates algorithms into two groups: first, traditional machine learning, which conventionally works with manually engineered features; second, neural network-based algorithms, known as deep learning, which typically uses word embeddings to represent textual contents (Zhang et al., 2018c).

2.3.1 Traditional machine learning

There are a number of previous TCC studies using traditional machine learning algorithms, and popular ones include Logistic Regression, Naive Bayes, decision tree and Support Vector Machines (Dadvar et al., 2012; Chavan and Shylaja, 2015; Wester et al., 2016).

Logistic Regression is a special regression used to model the probability of a certain class. It leverages the logistic function to take any input data and outputs a value between zero and one, which can then be used to classify an instance by a cut-off value (Westreich et al., 2010). Therefore, it solves classification questions rather than regression questions.

Naive Bayes classifier calculates the probability of an event based on prior knowledge of conditions that might be related to the event. In the task of text classification, the Naive Bayes classifier computes the posterior probability of a class based on the distribution of features in general in the document. This works with the "bag of words" assumption (Aggarwal and Zhai, 2012b). In "bag of words", a sentence or document is represented as the pool of its words, disregarding grammar and word orders (Aggarwal and Zhai, 2012b).

A **decision tree classifier** consists of decision nodes and leaf nodes. Each decision node splits input data into two or more downstream nodes connected to it according to a single feature (Stein et al., 2005). The downstream nodes repeated this step according to another feature until passing the data to leaf nodes. Each leaf node indicates a class that is the result of an instance (Stein et al., 2005). In the context of text classification, each text document is the input data which will be assigned to a class, i.e., a leaf node, after passing through layers of decision nodes.

A **Support Vector Machines** model represents instances as points in a high-dimensional feature space where each dimension is a feature (Tong and Koller, 2001). The basic concept of Support Vector Machines is to find the optimal hyper-plane to separate training data points as much as possible according to their classes (Korde and Mahender, 2012). The points of two classes closest to the decision plane are called the support vector. The machine learning process is to maximize the margin between support vectors from different classes (Tong and Koller, 2001).

2.3.2 Neural network-based methods (deep learning)

Overview

A neural network is constructed by a collection of connected nodes called neurons, inspired by the neurons in a biological brain. Each connection transmits a "signal" from one neuron to another. A neuron that receives a "signal" can process it and then signal additional neurons connected to it. This process will be repeated and the signal will be iterated to achieve better results through the learning process

(Lusci et al., 2013).

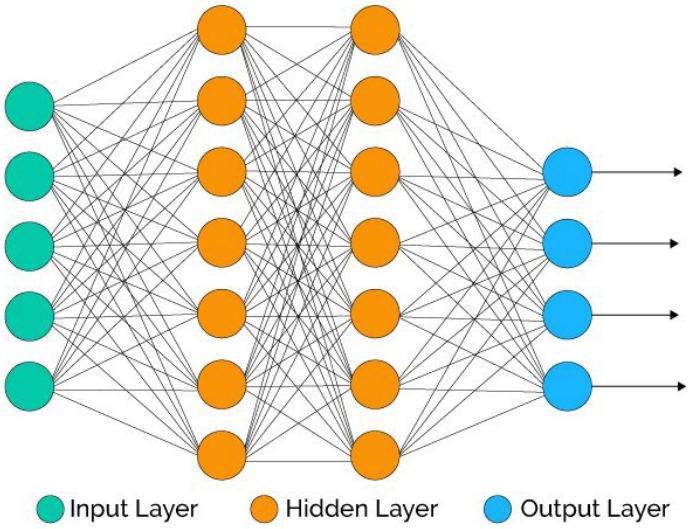

Figure 2.7: *A generic deep neural network (Florance, 2018). Neurons are represented by dots.*

Each iteration can be explained with two phases. The first is "forward pass" where the model receives the input signals and the input signals then traverse the network and leave "information" to neurons until reaching the last level of neurons (from left to right in Figure 2.7). The second is "backward pass" (from right to left) where the "information" is adjusted according to the "errors" uncovered during the forward pass. The adjustment is guided by the loss function which calculates the difference between the current "prediction" and the "golden standard". In other words, the loss function monitors the degree of "error" of the current model's prediction and guide the model (i.e., parameters in neurons) to adjust for better prediction. There are many different loss functions that calculate the "error" level differently. Commonly-used loss functions include mean absolute error, mean squared error, cross-entropy loss, negative log likelihood loss and Gaussian negative log likelihood loss (Paszke et al., 2019; Ruder, 2019; Wolf et al., 2019).

The iteration process is repeated, enabling the model to "learn" features. This is much akin to the process of how we humans learn from our mistakes and revise future judgment accordingly. The iteration process is the learning process. In the context of neural network-based algorithms, the "signals" and "information" consist of weighted values (usually in the form of a vector or matrix) and functions (Lusci et al., 2013). These weights and functions are often mentioned as *parameters*. Word embeddings discussed in Section 2.2.2 is an example of "signals".

Neural network models are conventionally constructed in a layered architecture.

They commonly contain neurons in three interconnected conceptual layers: input layer, hidden layer and output layer. Input layers encode data to a form that can be consumed by the neural network models such as vectors and matrix. Output layers take the final feature vectors learned from the penultimate layer (the last layer of hidden layers) and apply functions, like Sigmoid and Softmax, to output the probability distribution over labels (Lepe and Vázquez, 2017; Zhang et al., 2018b).

The hidden layers are between the input and output layers. These undergo the iterative training and learn features, as shown in the Figure 2.7. When a neural network consists of multiple hidden layers, it is called a deep neural network (DNN) or deep learning (Deng, 2014; Zhang et al., 2018b). In DNN, layers close to the input layer are conventionally called "lower layers". Lower layers are thought to learn superficial features like the shape and color of an image in computer vision or the general grammar of a sentence in NLP (Day and Khoshgoftaar, 2017; Tan et al., 2018). In contrast, layers close to the output layer are the higher layers and are thought to learn higher level features (Day and Khoshgoftaar, 2017; Tan et al., 2018).

When training a DNN, instead of feeding all data instances into a DNN model at once, only a small sample of data instances will be fed into the DNN every time. One sample collection of the data is called a *batch*. DNN processes the data batch by batch. It is conventional to use batch sizes as a power of 2[2], such as 32, 64, 128, 256. Conventionally, each data instance will be learned by a DNN more than once. The number of times a data instance has been used to train a DNN is called an *epoch*. In other words, for an epoch, DNN will work through the entire training dataset once. Various settings of epochs have been used and there is no general rule of thumb of an "optimal" number to use in research. Although, several empirically practices of setting up epoch numbers are commonly used which follow a similar principle that stop training when the model performance no longer improves (Deng, 2014; Aggarwal and Zhai, 2012a; Peters et al., 2018b; Founta et al., 2019; Kennedy et al., 2020; Zhao et al., 2021).

Parameters such as batch size and epoch times are called hyper-parameters. They are a crucial component for a DNN as they directly control the behavior of the training algorithm. Thus hyper-parameters have an important impact on the performance of the model during training. Currently, considerable skill and

[2]Using a power of 2 number is for the alignment of the virtual processors (VP) onto the physical processors (PP) of the GPU. Since the number of physical processors is often a power of 2, using a number of virtual processors different from a power of 2 leads to poor performance and a waste of virtual processors.

experience is required to select sensible values for hyper-parameters (Deng, 2014). Furthermore, sensible values for one hyper-parameter may depend on the values chosen for another. However, hyper-parameter tuning especially for DNN is computationally expensive (Deng, 2014).

Other important hyper-parameters include the number of hidden layers and the number of hidden units (how many neurons comprise a layer). For different DNNs, additional or alternative hyper-parameters may be required.

Overfitting is another essential concept in machine learning, which refers to the unfavorable learning outcome of a model when the model corresponds too closely or exactly to a particular dataset but fails to predict the other unseen dataset, i.e., the model memorizes the irrelevant noise instead of learning the signal and, therefore, performs less well on a subsequent new dataset (Yamashita et al., 2018). In the context of DNN, there are techniques to prevent or mitigate overfitting, such as dropout which cut out some neurons randomly; and regularization which adds a penalty in the loss function to discourage the complexity of a model.

Neural network-based methods have proven to be effective in the tasks of NLP (Goldberg, 2016). Many different DNNs have been explored in NLP. The **Feed-forward Neural Networks (FNNs)** was the first and simplest type of neural network devised (Schmidhuber, 2015). In this network, signals moves in only one direction, forward, from the input neurons, through the hidden neurons and to the output neurons. There are no cycles or loops in the network. A range of more complex DNNs has been developed to make neural networks more powerful. They distinguish from each other by their internal connection architecture, configuration and learning mechanism (Chen et al., 2017). Popular DNNs for text classification include Convolutional Neural networks, Recurrent Neural Networks and Neural Networks with Attention (Zhang et al., 2018c; Yang et al., 2016).

Convolutional neural networks

Convolutional Neural Networks (CNNs) were originally developed for image processing. They are inspired by the animal visual cortex where individual neurons respond to stimuli only in a restricted region (Matsugu et al., 2003). When recognizing an image, a CNN will "look at" a fixed sized region (sub-region) of the image and interpret the sub-region as a higher-level abstract concept such as a shape, or colour. In the context of processing text, a CNN will "look at" a certain length of characters or words (sub-region) in each sentence and transform each sub-region into one new abstract feature. How CNN extracts features from

each sub-region will be decided by the CNN filter, whose size is equal to each sub-region's size.

We use Figure 2.8 as a simplified example to describe how CNN works. In this example, we adopt 5-dimensional[3] word embeddings as features and concatenate them to represent a 7-word sentence. This leads to a 7x5 sentence matrix. We define a CNN filter represented by the 2x5 yellow matrix in Figure 2.8. First, the CNN filter overlays across the word vectors of "It" and "is" and performs an element-wise product and then sum them up. This outputs a number, 0.51. Then, the CNN filter moves down one word and overlays across the word vectors of "is" and "a" and perform the same operation to get 0.53. This process will be repeated until "be" and "islamophobia" have been "looked at" by the CNN filter. To capture latent features in various lengths of characters or words, many papers use multiple sizes of filters (Zhang and Wallace, 2015; Lepe and Vázquez, 2017; Zhang et al., 2018b). The filter size and the number of filters are both hype-parameters of CNN models (Yamashita et al., 2018).

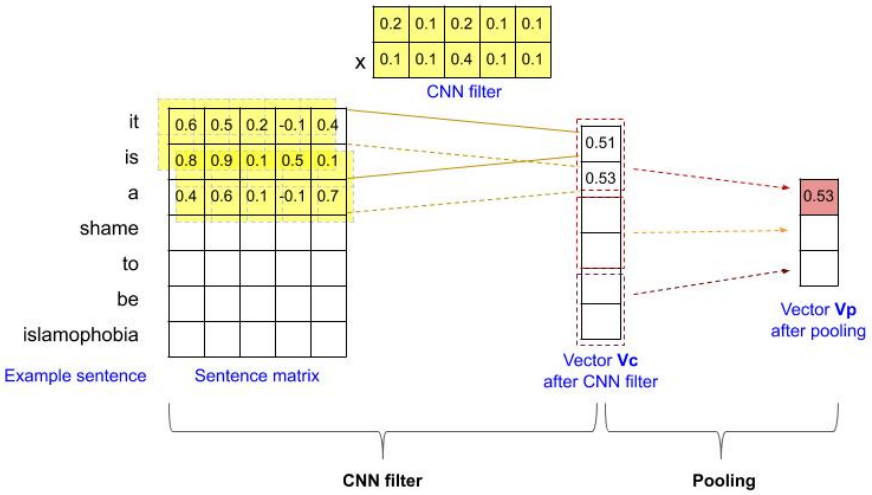

Figure 2.8: *A simplified illustration of how a CNN filter reads sub-regions one by one with stride size =1 and how max pooling works subsequently.*

CNN based models often utilise the *pooling layer* to further extract key features and condense the dimensionality of feature representations. Commonly-used pooling schemes are max-pooling and average pooling (Kim, 2014). They take the maximum and the average value of each sub-region respectively (Kim, 2014). As

[3]As mentioned in Section 2.2.2, in real word embeddings, the dimension size is often bigger than 100. We use 5-dimensional word embeddings to simplify the explanation.

shown in Figure 2.8, we obtain a number, 0.53, for the first two elements in V_C when adopting max-pooling of size 2.

CNN filters enable CNN models to extract features in context and capture 'local' information such as semantic clues (Che et al., 2017; Zhang et al., 2018b; Agrawal and Awekar, 2018). Due to this superiority, researchers have successfully used CNN in various toxic comment detection tasks. Badjatiya et al. (2017) have employed CNNs to classify tweets into three groups: racist, sexist and neither. This method has significantly outperformed the traditional methods such as LR and SVM. Gambäck and Sikdar (2017) have also investigated CNNs to detect hate speech. They attempt to improve performance by adding character-level features into Word2Vec, but discovered that CNNs with basic Word2Vec alone worked best. In the task of abusive comment classification, Chu et al. (2016) have explored CNNs featured with either word embeddings or character embeddings. They found using CNNs trained with character embeddings alone reached higher performance than word embeddings alone.

Recurrent neural networks

The central idea of **Recurrent Neural Networks (RNNs)** is to utilize the previous inputs and computations. CNNs assume that all inputs/sub-regions are independent of each other, while RNNs perform the same task for every element of a sequence with the output being dependent on the previous computation (Ordóñez and Roggen, 2016; Che et al., 2017). Accordingly, rather than processing all elements of a data at the same time (as shown on the left of Figure 2.9), RNNs process one element first and then process another element with the result of the previous element. As noted in Figure 2.9, the non-RNN model on the left processes data elements, X_1, X_2 and X_3 at the same time (time step t_1), while the RNN model processes X_1 first at time step t_1 and then process X_2 and the output from X_1 at t_2, subsequently X_3 will be processed at t_3 with the output from X_2 which contains the result information from X_1. In the context of NLP, a character, a word, or a phrase is from an 'ordered sequence', i.e., sentences. Therefore, they can be taken as the equivalents of ordered X_1, X_2 and X_3 in Figure 2.9. In this way, a time step is associated with a character, a word, or a phrase, depending on the setup.

An intuitive understanding of RNN is to think it processes each element in a data instance one by one and it has a "memory" of what has been processed previously. In the context of text processing, RNNs "remember" and "assess"

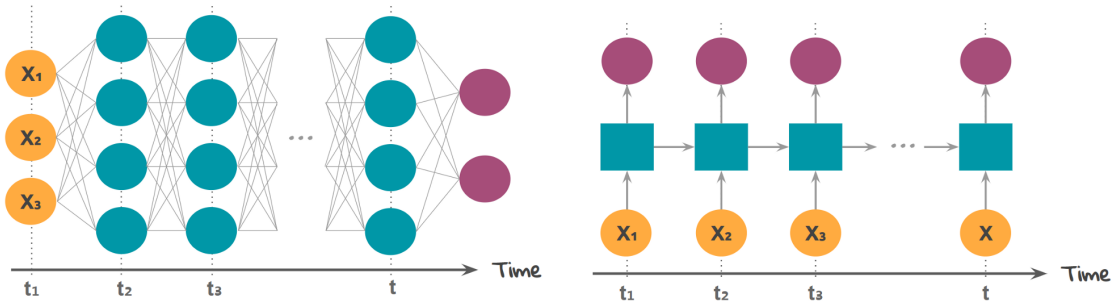

Figure 2.9: *An comparison of data processing methods between non-RNN neural networks and RNN (Jiwon, 2019).*

the context when trying to interpret the current word (Ruder, 2019). In theory, RNNs can utilize information in arbitrarily long sequences, but in practice, they are limited to looking back only a few steps. Information from long distances before will vanish in the later time steps and the information from the close distances will dominate. This can be compared to the situation that we will remember more clearly what happened in the last minute and forget what happened a week or a month ago.

Long Short-Term Memory network (LSTM) is a modified RNN model which is good at learning long-term dependencies (Chiu and Nichols, 2016). It is built on top of RNNs with additional functions of "forgetting" less important information and "remembering" useful information for later time steps. Like RNN, LSTM processes sequence data in order following a direction. The vanilla LSTM "reads" text from left to right, just like we humans. To further capture the sequence information of words and word dependency, Graves and Schmidhuber (2005) proposed bidirectional LSTM (bi-LSTM) which can "read" text from not only left to right but also right to left. Therefore, bi-LSTM captures both the previous and following contexts of a word.

In general, CNNs are able to learn the information from surrounding words or characters (i.e., the sub-regions as mentioned), but is weak at capturing the information of long-distance dependency in texts (Wang et al., 2016a). LSTM on the other hand, addresses this limitation by modelling texts across sentences in sequence for its "memory" function (Wang et al., 2016a).

A slight variation of LSTM is the **Gated Recurrent Unit (GRU)** (Cho et al., 2014). It combines the "forgetting" function and current word information into a single updating function. The resulting model is simpler and faster to compute.

In short, RNNs and their variations are good at learning orderly information

(Zhang et al., 2018c). Their ability to use internal memory to process arbitrary sequences of input has been found to be effective for text classification tasks (Johnson and Zhang, 2016; Agrawal and Awekar, 2018). Researchers have attempted to utilise RNNs to classify toxic comments. Del Vigna et al. (2017) employs a bidirectional LSTM fed with two word embeddings (concatenating two different word embeddings)[4] to detect hate speech in Italian. They have found that the accuracy of their LSTM model was higher when using coarse-grained binary labels ("hate" and "no hate"), compared to using fine-grained labels ("strong hate", "weak hate" and "no hate"). As discussed in the CNN section, character level features have also been explored on RNN models. In a binary classification task with comments being abusive or not, Mehdad and Tetreault (2016) have implemented a RNN using n-gram (n = 1...5) characters[5] as inputs instead of words. This approach achieves an increase of approximately 8% in average class accuracy.

Beyond features, researchers have attempted to improve their model's performance by combining different DNNs. Zhang et al. (2018c) proposes a convolution-GRU based deep neural network by incorporating a GRU layer on top of a convolutional layer to detect hate speech. They hypothesise that the CNN+RNN structure will be more effective as it will be able to capture co-occurring word n-grams as useful patterns for classification (Zhang et al., 2018c). For example, for the sentence "*These muslim refugees are not welcome in my Country they should all be deported ...*", such pairs include "*muslim refugees, deported*" and "*muslim refugees, not welcome*" (Zhang et al., 2018c).

Attention mechanism in deep learning

When inspecting objects, we humans usually look at certain specific parts with more attention depending on the context. For example, when we see a photo of a group of people, we tend to pay more attention to everyone's face if we want to find out who is in the group. Similarly, we will pay more attention to the environment if we want to know where the photo is taken. This is the intuition behind the "attention" mechanism in deep learning. Attention mechanism is first proposed by Mnih et al. (2014) to capture the important regions in image classification tasks. They added a "attention layer" on top of RNN layers. The attention layer learns which area in the image is more informative to the final classification prediction.

[4]The author have not specified which are these two word embeddings lexicons

[5]Different from the character embeddings in Chu et al. (2016)'s model mentioned in the last section on CNN, Mehdad and Tetreault (2016) adopted n-gram representations, one to five gram, which can be taken as bag-of-character model.

Then the corresponding features that RNN layers have learned will be given more weights. Intuitively, the model learns to "pay more attention" to the important regions in images.

Yang et al. (2016) were the first to apply the attention mechanism on text classification tasks. Intuitively, it is natural to compare looking at an image to reading text as we also pay different levels of attention to different words or sentences during reading. However, texts have a hierarchical structure, where words form a sentence and sentences form a paragraph or a document (Yang et al., 2016). Therefore, Yang et al. (2016) include two levels of attention mechanisms in their model, known as **Hierarchical Attention Networks (HAN)** (Yang et al., 2016).

The overall architecture of the HAN is shown in Figure 2.10. It consists of four main parts that serve two different attention mechanisms. First, the bidirectional GRU on the bottom (also labelled as word encoder) encodes the input to word embeddings initially. Second, the word attention layer extracts words that are important to the meaning of the sentence and aggregate the representation of those informative words to form a sentence representation. Following this, another bidirectional GRU (also labelled as sentence encoder) takes the input representing each sentence. Fourth, a sentence attention layer forms a document representation for the whole text that rewards sentences which are clues to correctly classify a document (Yang et al., 2016). The document representation then will be used as features for text classification.

In 2017, Vaswani et al. (2017) propose **Transformer**, which heavily uses attention mechanism. RNNs capture the orderly information and long dependencies between words by "remembering" what has been processed before. Different from the mechanism of RNNs, Transformer learns these information entirely relying on the attention mechanism and it has achieved SOTA performance on a range of NLP tasks. Many current SOTA DNN models are developed on top of Transformer (Radford et al., 2018; Devlin et al., 2018; Lample and Conneau, 2019; Liu et al., 2019b).

Transformer is composed of encoders and decoders[6]. Each encoder and decoder can be broken down into two sub-layers: the feed forward layer (in blue) and the attention layer (in orange) as shown in Figure 2.11 (Alammar, 2018). Transformer can also be simply taken as a FNN with full attention (Vaswani et al., 2017).

This configuration allows the transformer to learn an attention distribution for

[6]Recall that encoders and decoders are mentioned in Section 2.2.2 to explain the translation model used by CoVe.

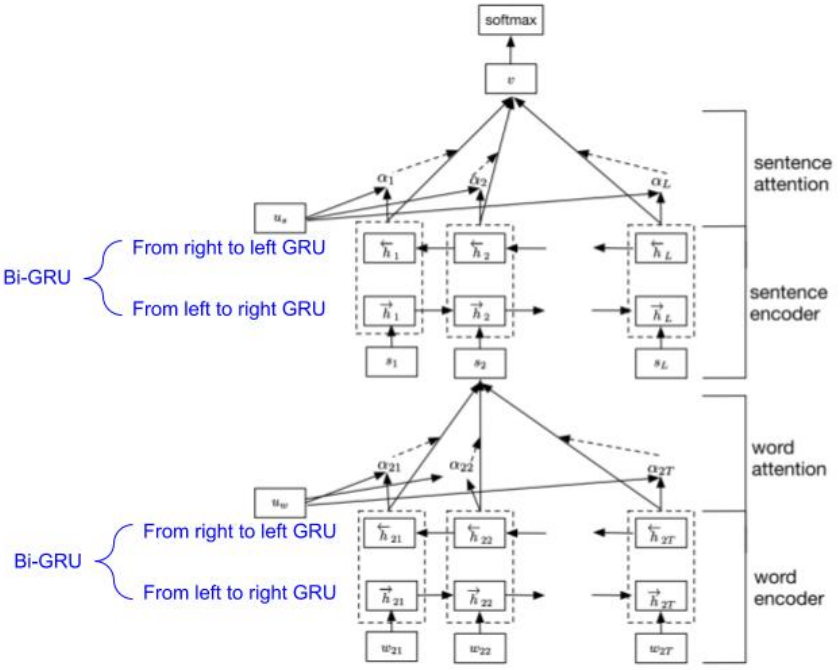

Figure 2.10: *Hierarchical Attention Network (Yang et al., 2016).*

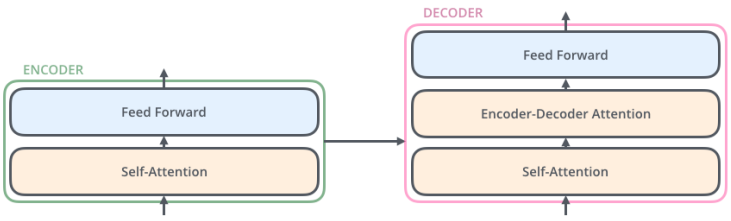

Figure 2.11: *A broken down of one single encoder and decoder in Transformer (Alammar, 2018)*

each input word over the entire input (Lukovnikov et al., 2019; Brunner et al., 2019). In other words, the attention distribution indicates how strong the input word relates to each word in the rest of entire input text as shown in Figure 2.12. For example, the attention between "law" and "its" is very sharp. This is because "its" refers to the aforementioned "law" in the previous context.

In the tasks of cyberbully comments detection, Agrawal and Awekar (2018) have compared four different DNN models, namely CNN, LSTM, Bi-LSTM, Bi-LSTM with attention. They build these four DNNs which differ only in the neural architecture layer (highlighted in yellow) as shown in Figure 2.13 while the remainder of the structures is the same (Agrawal and Awekar, 2018). The models are tested on three datasets with different labels built on Formspring, Twitter and

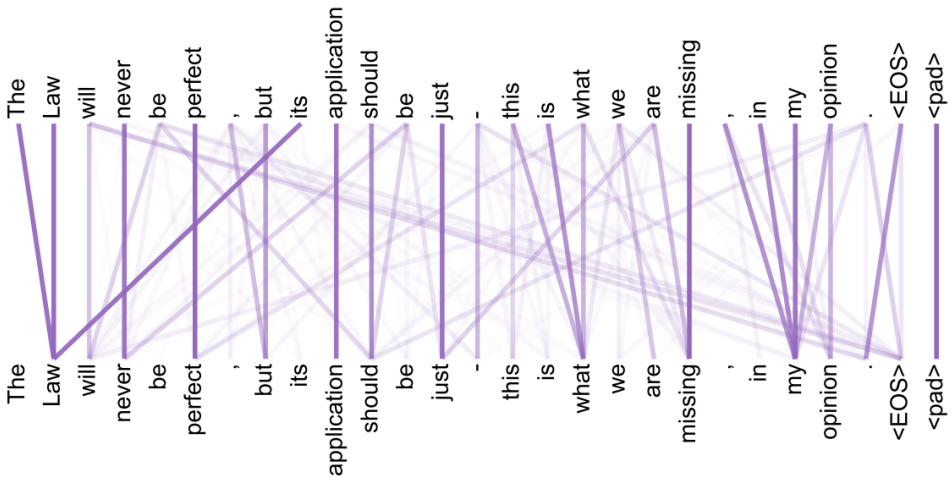

Figure 2.12: *The attention distribution captured by Transformer (Vaswani et al., 2017).*

Wikipedia. In the results, no models can consistently outperform others over all three datasets.

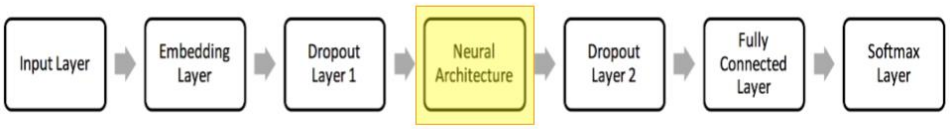

Figure 2.13: *The general model architecture of the four DNNs models in Agrawal and Awekar (2018)'s paper. The main difference between the fours (CNN, LSTM, Bi-LSTM, Bi-LSTM with attention) are the "Neural Architecture" as highlighted in the figure.*

Gao and Huang (2017) have explored Bi-LSTM with attention in hate speech classification and found that by adding attention mechanism to Bi-LSTM, the performance of the LSTM model improves by 5.7% in AUC score[7]. Pavlopoulos et al. (2017) also show that the attention mechanism improves the performance of the RNN model (GRU) when dealing with abusive comments in Greek from Greek sports news portal.

[7]AUC is a measure metric for classification problems which will be introduced in later sections

2.4 Transfer learning

2.4.1 Concepts

In the context of supervised text classification, for every task, a new model is trained from scratch on a labelled training dataset whose labels are assigned by human, i.e., "golden standard" labels. However, this manual labelling process is expensive and use-for-once labelling is inefficient. Another issue is that the appropriate size of the training set is difficult to determine. A dataset that is too small is potentially insufficient to train an effective model. For these reasons, it seems natural to seek methods that leverage other available datasets to train a model. One popular approach to do so is transfer learning (TL), a machine learning concept that transfers knowledge learned by one model to a second model targeting a separate task (Lu et al., 2015). The first task and its knowledge to be transferred are called the *source task*, the second task is usually referred to as the *target task* or downstream task (Ruder, 2019). The source task and downstream task do not need to share an identical label set. The source task do not even need labels in some TL cases, which will be discussed later in Section 2.4.2.

For example, assuming we have two annotated datasets available for two separate text classification tasks. One is product review data with tens of thousands of review comments labelled as either "positive", "negative" or "neutral". The other is a Facebook dataset labelled as "racist" or "not racist", which only contains a few hundred comments for training. TL will allow for knowledge that has been learned from product reviews to be transferred to help with the task of racist content classification. Thus, the product review is the source task and racist content is the target task. From the source task, i.e., product review classification task, the machine learning algorithm can learn generic features of human language alongside other low-level features.

Although TL is not limited to neural network-based methods, more recent TL research have primarily employed deep learning methods. One practical reason encouraging the use of deep learning with transfer learning may be the rapid development of deep learning since 2006 (Erhan et al., 2010; Deng, 2014; Kim, 2014; Lu et al., 2015; Yang et al., 2016; Zhang et al., 2018b). In addition, numerous studies have demonstrated the superior performance of neural network-based methods compared to traditional machine learning methods (Erhan et al., 2010; Lu et al., 2015). Another possible explanation is the different feature representation issues that traditional machine learning methods with manually engineered

features have to handle during TL. That is, the manually engineered features are usually designed specifically for the specific task and thus are hard to transfer to different tasks. Therefore, this thesis will focus on TL with deep learning, also known as deep transfer learning (Tan et al., 2018). Unless stated otherwise, we discuss each concept and method based on the use of DNN in the remainder of this thesis.

The process of TL typically consists of two stages, pre-training and adaptation (Ruder, 2019). In the first stage, the model is pre-trained on the source task. In the second stage, the pre-trained model is adapted/transferred to a downstream task and trained on the separate training data (Ruder, 2019). TL methods can be distinguished from each other by different pre-training methods and adaptation techniques, which will be discussed in the forthcoming sections.

2.4.2 Pre-training

In TL, what is transferred from the source task has a fundamental impact on the performance of the downstream task. Further, whether the pre-training needs labelled datasets or not also makes a big difference in the practicality of TL. Generally speaking, pre-training based on unlabelled datasets will have wider access to source data than those requiring labelled datasets. This "supervision condition" (i.e., using labelled or unlabelled datasets during pre-training) partially determines the difficulty and cost of TL. Therefore, the majority of literature separates different pre-training methods based on the supervision condition of pre-training (Ruder, 2019). This section discusses three TL pre-training methods according to the taxonomy proposed by (Ruder, 2019).

Supervised pre-training

In supervised pre-training, the model is pre-trained on another labelled dataset that is supposed to be a related task to the target task. In this way, creating new labelled data in the target task is replaced by using labelled data in the source task during pre-training. By making use of the extra data from a related task in the pre-training, supervised pre-training TL increases the labelled training data in general, allowing the classifier to learn from more labelled data. According to (Day and Khoshgoftaar, 2017), when the source task has very little in common to the target task, the transferred knowledge may harm the target model, also known as negative transfer. For this reason, in many cases only suitable existing tasks are chosen as source tasks for a particular target task (Raina et al., 2007). Having

said that, this also limits the practical implementation of supervised pre-training as it is not always easy to find a source task that is similar to the target task.

CoVe is one famous supervised pre-trained model as discussed in the previous Section 2.2.2. Rather than text classification, CoVe is pre-trained on an English-Germany translation task. The pre-training for the source task. i.e., English-German translation, uses a parallel corpus for the two languages. The paired German text for an English text that is to be translated to is the "golden standard" for supervised learning.

Distantly supervised pre-training

Distantly supervised pre-training is a learning scheme where the classifier is learned on a weakly labelled training set (Ruder, 2019). Here, data are labelled automatically based on heuristics or rules instead of human annotators (Ruder, 2019). This results in the training data having "loose standard" labels (also known as noisy labels) rather than "golden standard" labels like supervised pre-training does (Go et al., 2009).

A straightforward example, as well as the most common application is using emoticons included in a comment as the comment's label in sentiment analysis tasks (Go et al., 2009; Suttles and Ide, 2013; Felbo et al., 2017). In this way, the comments with the same emoticon will be categorised to one group under the shared emoticon and the model is supposed to predict the emoticon. For example, comments with smiling face will be classified into one group, and comments with angry face will be in another group. The training objective is to predict which emoticon the comment contains. Conventionally, researchers removed the emoticon from the comment and then trained the model to predict the emoticon in the interest of ensuring the classification relied solely on the language (Go et al., 2009; Suttles and Ide, 2013; Felbo et al., 2017). The intuition behind this was to take the emoticons as the underlying indicator of sentiments in the comments. This allows the model to learn the features of different sentiments. In addition to the "emoticon" labels, distant supervision may also use hashtags as "loose standard" labels for sentiment classifications (Purver and Battersby, 2012; Hasan et al., 2014; Silva et al., 2016). In addition to sentiment analysis tasks, distantly supervised pre-training have been employed in products review classification and movie reviews classification by taking star ratings as "loose standard" labels (Silva et al., 2016).

Undoubtedly, using emoticons, hashtags or star ratings as labels is not perfect

as one group could have some mis-classified comments, i.e., the noisy data, that do not fit in the comments group. For instance, a positive emoticon can indicate sarcasm or a compliment in an otherwise negative text, i.e., a false positive instance (Felbo et al., 2017). However, the advantage of this method is the wider availability of training data. Increasing training data may compensate for the noisy instances. The ideal algorithm will be robust to noisy data and able to learn from the larger proportion of true positives. The underlying information of emoticons and review ratings are usually closely related to TCC as the emoticons and review ratings indicate the sentiment that are commonly used as features for TCC tasks (Purver and Battersby, 2012).

Unsupervised pre-training

In contrast to supervised learning, unsupervised learning is a machine learning concept that does not require pre-existing labels for the training data. Instead of outputting predictions, the goal of unsupervised learning is to model the underlying structure or pattern in the data in order to learn more about the data. Two classic examples of unsupervised learning are clustering and dimension reduction (Ghahramani, 2003). Clustering groups data according to its features, identifying commonalities in the data and grouping data based on the presence or absence of such commonalities. Dimension reduction aims to reduce the number of features.

In the context of NLP tasks, language modelling is the most widespread unsupervised learning method (Howard and Ruder, 2018; Devlin et al., 2018). The basic idea of language modelling is to calculate the probability of sequences of words in a text, based on the appearing frequency of the sequences in the whole corpus. In other words, it estimates the relative likelihood of different phrases in a corpus, which is useful in many NLP applications. For example, if in a corpus, "United" and "Kingdom" appear together many times, language modelling will assign the sequence of "United Kingdom" a high probability. This can be used in autocomplete where the system will suggest "Kingdom" as the next word when a user inputs "United". Due to the mechanism that the target a language model aim to predict is in the raw input data, language modelling is also often referred to as "self-supervised learning" (Naseem et al., 2021). Language modelling has proved to be very effective to capture many facets of language, such as long-term dependencies, hierarchical relations, and sentiment (Linzen et al., 2016; Radford et al., 2017; Gulordava et al., 2018; Howard and Ruder, 2018).

Early language models are trained to predict the next token from either left-

to-right or right-to-left. The probability of sequences then will be iterated based on the prediction. Intuitively, it is reasonable to believe that a bidirectional model is more powerful than a unidirectional model (Devlin et al., 2018). Peters et al. (2018b) proposes a bidirectional language model that first independently trains a language model predicting tokens from-left-to-right and a language model predicting tokens from-right-to-left, and then concatenate these two language models. Also aiming for bidirectional pre-training, Devlin et al. (2018) first use masked language model in their pre-training. Masked language model randomly masks some part from the input and its objective is to predict the masked part on the context. As explained in the previous Section 2.2.2, Elmo is pre-trained on language modelling to generate contextualised word representation, which is also a form of unsupervised learning (Ruder, 2019). Pre-training a language model is widely used in TL and it demonstrates robust performances on a wide range of downstream tasks. Section 2.5 will introduce more different commonly-used pre-trained language models in details as they are core techniques used in this thesis.

2.4.3 Adaptation

The previous section focused on different pre-training approaches for the first stage of transfer learning. The second stage of "adaptation" transfers the knowledge learned from the source task to the target task. Devlin et al. (2018), Houlsby et al. (2019) and Ruder (2019) all categorise adaptation techniques into two groups: features-based transfer and fine-tuning. Features-based transfer is also referred to as feature extraction by Ruder (2019). In feature-based adaptation, the pre-trained model is taken as the input, fed into the downstream model. As the name suggests, the role of the pre-trained model is to build feature representations of input data. In contrast, fine-tuning adapts a pre-trained model directly and trains it on the target task (Ruder, 2019). A TL model can use either or both of the two techniques.

Feature-based transfer

In feature-based TL, features are transferred as fixed parameters and weights in transferred layers, known as feature representations or representations, from the source task to the target task. A task-specific DNN model takes the representations, i.e., pre-trained layers, and is trained on the target task, where only the DNN model is trained from scratch. Here, the task-specific DNN model stacks on top of the transferred representations. The transferred model or representations

are fixed which will not be updated during the training after adaptation.

As shown in Section 2.2.2, Word2Vec and Glove can be taken as examples of feature-based adaption. They are both pre-trained on separate source tasks, i.e., language modelling. The representations learned from the pre-training are saved as fixed-vectors that are fed to different downstream tasks.

Fine-tuning

In contrast, fine-tuning TL adapts the source task model to the target model and updates the "knowledge" from the pre-trained model. In such a way, the pre-trained model forms part of the target task model, which may contain other layers that are configured to be task-specific and to be learned from scratch. However, their "learning" may be "influenced" by the transferred, pre-trained model. This can also be interpreted as replacing several layers in the model of source task with new trainable layers so that a new model for the target task is built. The training process of the new model is a fine-tuning process for the pre-trained part (i.e., layers of the pre-trained model) and a learning-from-scratch process for the new added layers. This allows the model to take advantage of the knowledge learned from pre-training and also update the knowledge based on the new task. However, the updating mechanism of fine-tuning has the risk of forgetting about what features have been learned from pre-training, a challenge termed catastrophic forgetting (French, 1999; Schmidhuber, 2015; Houlsby et al., 2019).

Different fine-tuning approaches have been developed to optimise the TL effectiveness and to avoid catastrophic forgetting. Universal Language Model Fine-tuning (ULMFiT) is one of the most popular methods on fine-tuning on NLP tasks proposed by (Howard and Ruder, 2018). ULMFiT uses a gradual unfreezing manner to fine-tune the model (Howard and Ruder, 2018). It first unfreezes the last layer and fine-tunes the model for one epoch. In this step, only one layer has been updated. Next, ULMFiT unfreezes the adjacent lower layer and fine-tunes all unfrozen layers for one epoch. This repeats until all layers are unfrozen and fine-tuned.

2.4.4 Related work in TTC

There are a number of studies on toxic comments classification using transfer learning in the last few years, along with the course of this PhD study.

Risch et al. (2018) have employed TL in a supervised pre-training manner in an offensive language detecting task. According to Risch et al. (2018), since

the target dataset only has 5,000 instances which do not work well with typical deep learning methods, they pre-train their model on a pre-existing labelled toxic comments dataset of 150,000 instances. The given target task aims to assign text into four categories: profanity, insult, abuse and other, different from the source task which has six labels: toxic, severe toxic, obscene, threat, insult and identity hate [8]. Wiedemann et al. (2018) believe the target task is similar to the source task and therefore, they keep the first layer frozen to include general features from the source task and add new layers on top of it to learn the target task. This can be taken as supervised pre-training and feature adaptation TL.

In Agrawal and Awekar (2018)'s study, three cyberbullying classification datasets from different social media platforms have been pre-trained for each other to check if the knowledge gained from one dataset can be used to improve cyberbullying detection performance on other datasets.

In their abusive content detection, Uban and Dinu (2019) pre-train the model using distantly supervised learning, where content in the source dataset were automatically labelled as positive or negative based on emotions. The automated labelling process assumes that any tweet with positive emoticons, such as ":)", are positive, and tweets with negative emoticons, such as ":(", are negative. Next, the pre-trained model has been transferred to two separate tasks, aggressive language detection (12k comments from Facebook) and offensive language detection (14k Tweets), to study the effect of transferring knowledge across tasks. Different adaptation methods also have been tested in their experiments. First, they have used fine-tuning to transfer features which either allows the model to update all parameters in all layers or only update the rest layers except embedding layers (Uban and Dinu, 2019). Second, they have tried feature-based transfer by only sharing the first embedding layers (Uban and Dinu, 2019).

Aside from supervised pre-training and distant learning, unsupervised pre-training have been tested in several studies of toxic comments detection. Korzeniowski et al. (2019) pre-train their model on PolEval, an unlabelled dataset in Polish, using language modelling and then transfer it to the hate speech classification task. They choose to adapt source task by ULMFiT fine-tuning. According to their results, ULMFiT significantly improved the model performance.

In Wiedemann et al. (2018)'s research of offensive language detection, they compare three pre-training methods. These are: 1. supervised pre-training on dataset of detecting "inappropriate" and "discriminating" content, which can be taken as a near-offensive classification task; 2. distantly supervised pre-training on

[8]The "clean" comments will not be assigned any label of these six labels

tweets labelled with emoticons they contain; 3. pre-training by Latent Dirichlet Allocation (LDA), an unsupervised learning algorithm that represents a text with a topic distribution where each topic is defined with a word distribution. They also have explored different fine-tuning methods for TL. Similar to ULMFiT, they gradually unfreeze layers in the fine-tuning but tried with different unfreezing orders. In their results, the unsupervised pre-training TL outperforms the other two, while none of the fine-tuning methods demonstrate a significant advantage.

2.5 Pre-trained language model (PLM)

Language model (LM) has been briefly introduced in Section 2.4.2. It is an an unsupervised learning method that aims to model the probability distribution over sequences of words or tokens. Pre-trained language models (PLMs) are language models pre-trained on source tasks and can be transferred to downstream tasks, especially to supervised tasks. This section will discuss PLM in detail as it is widely-used on a range of text classification tasks these years and several new SOTA NLP models are built upon it. Also, it is an important component for many TL applications. We do not aim to cover all fundamentals in PLM but the essential concepts and techniques of PLM that are relevant to or used in the research of this thesis. All concepts and techniques discussed are therefore in the context of using DNN. We first introduce different pre-training language models from the perspective of their pre-training tasks. Then we cover the popular PLM used in NLP. Last, we discuss the studies of applying PLM to TCC tasks.

2.5.1 Training tasks for PLMs

In the context of neural network based language models, a classic language model is the neural probabilistic language model proposed by Bengio et al. (2003), also known as **statistical language model**, which has been applied in many related studies (Mikolov et al., 2009, 2013; Han et al., 2021). Similar to traditional language model, it aims to model the conditional probability of the next word given all the previous ones. The two versions of LM used by Word2Vec (as introduced in Section 2.2.2 and illustrated in Figure 2.5), Continuous Bag-of-Words Model and Continuous Skip-gram Model, are variants built upon on traditional language models. One main difference is that Word2Vec takes advantage of the context before and after a given word rather than only words before it.

Masked language modelling (MLM) is a fundamental language modelling

method more widely used in recent years. It is first adapted by Devlin et al. (2019) in the context of DNN. The main idea behind it is to mask out some random word tokens in the input sentences and then train the model to predict the masked tokens based on the remaining tokens in the sequence. Intuitively, MLM is efficiently trained as it only masks and predicts a part of tokens in the sequence but all tokens interact as they are the context.

Two limitations of MLM have been identified by Yang et al. (2019). First, MLM corrupts the input with masks. That is, the masked tokens appear in pre-training but not in fine-tuning. Second, it neglects the dependency between masked positions. For example, given a sentence "New York is a city", if MLM masks "New" and "York", then the training objective would be:
$\log p(New \mid is\ a\ city) + \log p(York \mid is\ a\ city)$.
In such a way, the model fails to capture the dependency between "New" and "York". To handle such limitations, Yang et al. (2019) proposed **permuted language modelling**. Permuted language modelling is trained to predict a token on all possible permutation of the remaining tokens in the input sequence. Given the same example above, Permuted language modelling predicts "New" based on the remaining tokens, i.e., "York", "is", "a" and "city", in different permutations[9]. Permuted language modelling aims to capture the high-order and long-range dependency of words in sentences via the different permutations.

Translation language modeling (TLM) proposed by Lample and Conneau (2019) is an extension of MLM, which is trained on pairs of text in different languages. Its training objective is, as MLM, to predict masked tokens based on context in different languages. Take English and French as an example, given a pair of text in English and French (as known as parallel text) where they are the translated version to each other, the pair of text is first concatenated. For example, the pair of "Hello" and "Bonjour" will become "Hello Bonjour". Then the model randomly masks some tokens and predicts those masked tokens as MLM does. The intuition is to allow the model to learn from the context in both languages. Especially when the context in the language of the masked one is not sufficient for the model to infer, the model can leverage the context from the translated text.

[9]Note that, permuted language modelling only permutes the factorization order (tokens representations in the embeddings) to reflect the different permutations of tokens, not the sequence order.

2.5.2 Popular pre-trained language models

BERT is one of the most widely-used PLMs. It has been commonly adopted in various NLP tasks as it has achieved SOTA results on 11 NLP tasks and it is able to generalise over a wide range of tasks (Devlin et al., 2019). A rich body of new techniques and applications are built upon BERT (Huang et al., 2019; Beltagy et al., 2019; Liu et al., 2019c; Sanh et al., 2019; Han et al., 2021; Wu and Ong, 2021).

BERT is trained on two tasks: MLM as discussed above, and predicting whether two sentences follow each other. For BERT, its MLM process masks 15% of words in the input sequence. The MLM objective enables the model to learn a representation based on its previous and following context. Its pre-training corpus includes BooksCorpus (800 million words) and English Wikipedia (2,500 million words), with a total size of up to 16GB (Devlin et al., 2019). This very large size of training data allows BERT to learn not only the superficial features of human language such as grammar and part-of-speech but also the semantics. Figure 2.14 illustrates the architecture of BERT in the context of its application on text classification tasks.

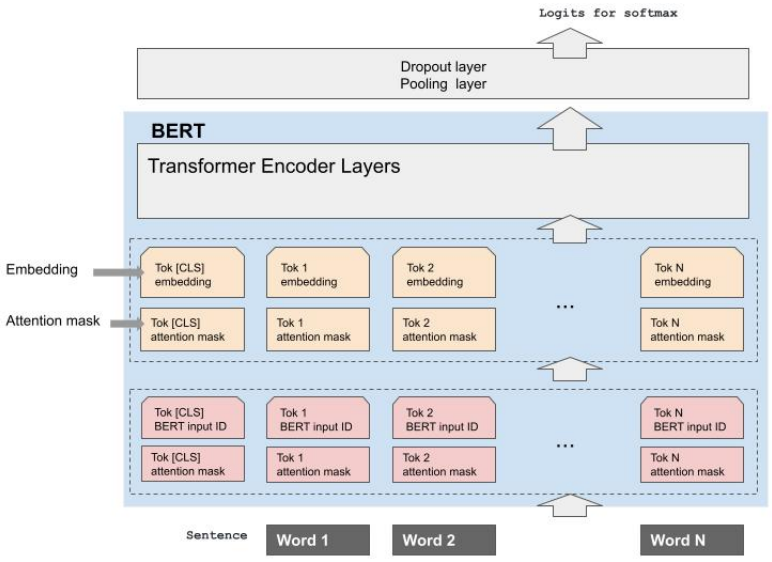

Figure 2.14: *Illustrations of a BERT model on classification tasks. This figure is better viewed in colour.*

As shown in Figure 2.14, BERT tokenizes sentences with WordPiece where sentences are splits into words and some words are further split into subwords. For example, the word "unhappy" will be split to 2 tokens of "un" and "happy". Using subwords has the advantage of reducing out-of-vocabulary tokens (Gillioz

et al., 2020). BERT takes token sequences of fixed lengths where longer sequences are truncated and shorter sequence are padded. Given a fixed length token sequence, BERT assigns each token an input ID (which is associated with a initial embedding) and an attention mask as coloured in pink in Figure 2.14. The embeddings are learnt from the language model pre-training. The attention mask indicates if the token is a padded token or not to avoid performing attention on padded tokens. The attention mask value takes either 0 or 1, where 0 indicates the token is a padded token for the fixed length so it will not be attended and 1 indicates the token is an actual token from the input text so it will be attended. The main architecture of BERT is built on Transformers as discussed in Section 2.3.2. This includes 12 layers of bidirectional Transformer encoders. We show these transformer encoders in grey in Figure 2.14. Embeddings and masks are fed into the Transformer Encoder layers and the final output is passed to the pooling and dropout layers for the final loss calculation with cross entropy in the training and for the classification with softmax in the evaluation. Pre-trained BERT models are published in two main versions, BERT base and BERT large. BERT base has a total of 12 attention heads and 110 million parameters. BERT large has 16 attention heads with 340 million parameters. BERT base has 768 hidden layers whereas BERT large has 1024 hidden layers. Also, there are BERT versions for different languages, such as Polish, Chinese, German and so on (Han et al., 2021; Kłeczek, 2020; Risch et al., 2019; Cui et al., 2021).

Many later PLMs are built on BERT with modified training tasks or architectures to improve BERT. These models include, for example, RoBERTa, XLM, XLNet, BART and CamemBERT (Liu et al., 2019c; Lample and Conneau, 2019; Yang et al., 2019; Lewis et al., 2019; Martin et al., 2019). We briefly introduce some as examples. RoBERTa uses a similar transformer-based architecture as BERT but removes the Next Sentence Prediction task from BERT's pre-training and modifies the MLM with dynamic masking (Liu et al., 2019c). By dynamic masking, the masked token changes during the training epochs. In other words, in each training epoch, the masked tokens are different. This is different from BERT that masks fixed tokens for each training epoch. What is more, RoBERTa enlarges the pre-training corpus 10 times bigger. XLNet is another PLM built upon Transformer. Different from BERT, it trains on permutation language modelling (Yang et al., 2019). It also uses a much bigger pre-training corpus of 130GB (Yang et al., 2019).

A few studies have paid particular attention to minimising the model size (i.e., total parameters) and speeding up the training process but maintaining com-

parable or better model performances. For example, DistilBERT is trained to reproduce the behaviour of BERT and it manages to achieve comparable performance to BERT on the language understanding benchmark GLUE by using just half the number of parameters of BERT base version (Sanh et al., 2019). ALBERT have further reduced the parameters to just about 11% of the BERT-base model (Lan et al., 2019). By two designs for allocating the model's capacity more efficiently, ALBERT largely reduces the parameter numbers and also maintained comparable performances across the benchmarks. First, the embedding matrix is split between input-level embeddings with a relatively-low dimension (e.g., 128), while the hidden-layer embeddings still use high dimension representations (768 as in the BERT case, or more). Second, the authors have observed that the network often learned to perform similar operations at various layers, using different parameters of the network (Lan et al., 2019). This redundancy is eliminated in ALBERT by parameter-sharing across the layers, i.e., the same layer is applied on top of each other.

Some research explores continued pre-training of the PLM using a large in-domain corpus to tailor a PLM to tasks in specific domains. In other words, continued pre-training of a PLM takes a pre-trained LM and continue pre-training it on the unsupervised language modelling task with an in-domain corpus. As a result, the model for the downstream task is initialized with weights from a further fine-tuned LM. Such further pre-trained PLMs include SciBERT for scientific papers, ClinicalBERT for clinical notes and predicting hospital re-admission, BioBERT for biomedical documents, FinBERT for financial sentiment analysis and TweetBERT for Twitter text (Beltagy et al., 2019; Huang et al., 2019; Lee et al., 2020; Araci, 2019; Qudar and Mago, 2020). Rather than for tailoring to a specific domain, Gururangan et al. (2020) proposes task-adaptive pre-training (TAPT) for tailoring to a specific task or dataset. TAPT uses unlabelled training data from downstream target task to further fine-tune RoBERTa. That is, labels of the target task datasets are removed and this "unabelled" dataset is used to further pre-train the LM using unsupervised LM tasks. Hence, the data size for further fine-tuning LMs in domains has been largely reduced compared to SciBERT, ClinicalBERT, etc. However, in their proposed TAPT, significant computing resources are still required due to the high settings of hyper-parameters, particularly due to the batch size and the epoch.

2.5.3 Related work in TCC

As a sub-task of text classification, TCC follows a similar trend shift from traditional machine learning methods to DNN-based methods, and to PLM-based methods. Especially since 2019, a number of TCC studies have begun to use PLM.

A growing body of literature has investigated the PLM performances on non-English TCC tasks as usually the TCC training dataset in non-English are very limited. Some typical such works include Plaza-del Arco et al. (2021) for Spanish with BERT, XLM and BETO; Pham et al. (2020) for Vietnamese with RoBERTa; Banerjee et al. (2020) for Indian with RoBERTa, XLNet and DistilBERT; Farha and Magdy (2020) for Arabic with BERT; and Lavergne et al. (2020) for Italian with multilingual LM (mBERT and XLM-RoBERTa) and Italian LM (AlBERTo, PoliBERT and UmBERTo). Most of these works are empirical comparison studies that investigate whether PLMs perform better than non-transfer learning methods or compare different PLMs on TCC tasks in non-English language models. A few similar works on English TCC tasks emerged when BERT was published (Baratalipour et al., 2020).

A few studies on TCC have explored strategies on how to transfer PLMs effectively. Mozafari et al. (2019) transfer BERT to different downstream architectures on two hate speech classification tasks. In the shared task on aggressive content detection, TRAC-2, a few papers utilise pre-trained LMs but with a focus on improving the model performance on one specific TCC task rather than generalising their methods to multiple TCC tasks or pre-trained LM models (Kumar et al., 2020). Most of them use BERT directly without complex downstream neural-network or continued fine-tuning (Baruah et al., 2020; Samghabadi et al., 2020; Gordeev and Lykova, 2020; Liu et al., 2020). It is unclear how other pre-trained LMs, such as RoBERTa and XLM, perform and how generalizable they are on a wide range of TCC tasks.

2.6 Research gap

This chapter laid out the background knowledge of text classification and TCC. It also discussed two SOTA techniques, transfer learning and pre-trained language model that this thesis is built upon. In this part, we review several challenges that remain to be addressed in TCC and that are the focus of this PhD study.

2.6.1 Gap 1: Transferring PLM to TCC tasks

Several studies on TCC have illustrated the strong performance of pre-trained language models (Mozafari et al., 2019; Baruah et al., 2020; Samghabadi et al., 2020; Gordeev and Lykova, 2020; Liu et al., 2020). This can be linked to the fact that using pre-trained models on downstream tasks essentially increases the overall training data size. Our previous work has also identified the correlation between DNN models' performance and the training dataset size, especially the number of positive data (the number of toxic comments) (Zhao et al., 2019). In short, the more training data, the better performance TCC models can achieve[10]. Given the high cost of labelling training data, utilising pre-trained language models as a means to "transfer" additional knowledge to the task has its practical utility.

By the time of conducting this study, there were a limited number of studies on transferring pre-trained language models to TCC tasks, and most of them focused on binary TCC tasks. On the other hand, multi-class and multi-label classifications make the learning harder by nature, and different types of toxic comments could have some intersectionality, hence increasing the complexity of the problem. However, they are closer to the real-world scenarios (Fang et al., 2017; van Aken et al., 2018; Liu et al., 2019a). Also, it is difficult to conclude how to best make use of PLM for TCC tasks or how to transfer PLM to TCC tasks effectively. This is because, as discussed in Section 2.4.4 and Section 2.5.3, previous studies have investigated different TL techniques and different PLMs on various TCC tasks with disparate focus, making their results difficult to compare. In short, although a few TCC works have been done using PLM, there remains scope for us to further exploit all benefits of PLM-based methods for TCC tasks.

Due to the complexity of PLM-based models and methods, there are many perspectives from which this can be explored, such as downstream architectures, hyper parameter settings and adaptation methods. Chapter 3 will explore these different settings and components in PLM-based models.

2.6.2 Gap 2: Identity term bias found in TCC models

In the field of machine learning, a model or a system is considered to be "fair" when its outcomes are not discriminatory according to certain attributes, like gender, race or nationality (Garrido-Muñoz et al., 2021). On the contrary, a model demonstrates "unfairness" if there is a clear bias on how decisions are made depending

[10]Although this pattern starts fading out after the positive data increase to 400, in many cases, valid positive data of one class in a training dataset could be less than 400.

on the values of that certain feature. This is also referred to as unintended bias (Dixon et al., 2018). There have been many works that illustrate the vulnerability of data-driven NLP systems in unintended learning biases (Bolukbasi et al., 2016; Dixon et al., 2018). These studies have primarily focused on DNN-based methods. This is because, first, DNN-based methods are widely used in recent applications. In particular, PLM based methods have achieved SOTA performance on a wide range of tasks. However, PLMs are pre-trained on very large natural language corpora, which themselves may contain bias. As a result, PLMs can potentially learn such bias that may subsequently affect the downstream applications. This is different from traditional machine learning models that use manually-engineered features, which can be designed in a way to avoid bias.

In the context of TCC, a few previous papers have narrowed down their focus on gender bias, racial bias and dialect bias (Field et al., 2021). Typically, such bias is reflected in language as phrases or terms that characterise a specific group of people such as "Muslim" and "black". This is known as identity term bias. These works have investigated the identity term bias from different perspectives, such as debiasing the training dataset to mitigate such bias or debiasing the algorithms (Zhang et al., 2018a; Prost et al., 2019; Xia et al., 2020; Mozafari et al., 2020; Kennedy et al., 2020). In short, identity term bias in TCC models is a research area that has taken off in recent years. While despite increasing number of studies looking to address identity term bias in TCC models, we identify a new angle to study this problem: rather than looking at the imbalance distribution between instances with and without identity terms, or how the model attends to identity terms, we investigate the subjectivity of a comment with respect to identity terms. We explore this further in Chapter 4.

2.6.3 Gap 3: Debiasing the model to mitigate identity term bias

Current methods mitigate identity term bias from two perspectives: debiasing the dataset and debiasing the model (i.e., algorithms) (Mozafari et al., 2020; Wiegand et al., 2019; Dixon et al., 2018; Davidson et al., 2019). Methods of debiasing the dataset curate the data, such as changing the distribution of the dataset and adding adversarial data. Debiasing the model aims to design the model to "be aware of" identity term bias. However, debiasing the dataset is essentially modifying the dataset, and this has been seen as controversial in the NLP research community (Bender and Friedman, 2018; Bender et al., 2021; Buckman; Rogers,

2021).

On the other hand, methods of debiasing the model can be divided into two groups: using an ensemble model and adding regularisation terms. Intuitively, an ensemble model follows the idea that making one part of the model learn the bias-related features and then the other part learn the rest "bias-free" features. Using regularisation terms is to punish the model for using bias to make prediction. In short, these two methods attempt to handle identity term bias with a similar principle: encourage the TCC model to ignore or pay less attention to the identity terms. We argue that this overlooks the fact that identity terms can be essential and important features to make predictions. This thesis aims to explore a new approach, which will be presented in Chapter 5.

Chapter 3

Improving the effectiveness of transferring pre-trained language models for toxic comment classification tasks

As a mainstream approach, pre-trained language models (PLMs) have been explored in various NLP tasks, and they have approved their robust performances. There are different factors that affect a PLM's performance, and their effect might not be consistent over different tasks. A large number of studies have explored these factors to improve the performance of downstream tasks. However, these studies do not focus on TCC tasks but explore a more general range of NLP tasks such as Question Answering, reading comprehensive, generic text classification. By the time this study was completed, there was a very limited number of studies exploring how to best use PLMs on TCC tasks. Therefore, this study focuses on this area.

The remainder of this chapter is structured as follows. Section 3.1 introduces the research area of transferring a PLM to a downstream task, subsequently introduce our research questions and then highlight the key contributions of our work in this chapter; Section 3.2 reviews related work of using PLM on TCC tasks, which are divided into three groups corresponding to the three aspects we will investigate; Section 3.3 explains our experimental methods of transferring PLMs to downstream TCC tasks. Section 3.4 presents our experimental implementation details and results. Section 3.5 concludes the work and discusses future work.

3.1 Introduction

Previous research has found that the increase of training data usually brings improvements in model performance. For all that, more training data does not benefit the model learning proportionally to the effort of creating these data. In other words, creating an amount of additionally labelled training dataset and using it once for a limited performance improvement is not an effective method overall as there is a "diminishing return" after the training data increases to a certain size. For example, in our earlier paper on studying the impact of training dataset sizes for TCC, we have tested one traditional statistical machine learning model, Support Vector Machine, and two DNN-based methods, CNN and LSTM. For all three models, their performance increase slows down after the amount of training data reaches a certain point (Zhao et al., 2019). What is more, it is difficult to decide how much data is enough for a new task in advance. Therefore, labelling more data is not always an ideal method.

As mentioned in Chapter 2, to tackle the lack of training data in supervised learning, current SOTA has shifted the focus to transfer learning, and in particular, the use of PLMs as a way to transfer knowledge learned from separate unsupervised tasks on massive unlabelled training data. This can be taken as an alternative way to utilize more data overall. The basic idea is to extract the pre-trained neural network layers from the PLM and add new neural network layers (i.e., "downstream architectures") on top of them to tailor for the downstream task (Gururangan et al., 2020). In other words, features from the PLM are transferred to downstream architectures to build a model for the target classification task. Intuitively, the knowledge learned from the pre-training will be helpful for the downstream task, especially when the task has a limited training dataset.

Previous studies explore different approaches to transfer a PLM, such as using different downstream network architectures as discussed in Section 2.5.3 and continued pre-training a PLM using in-domain corpus as discussed in Section 2.5.2 (Kennedy et al., 2020; Mozafari et al., 2019). Within the area of TCC research, by the time this work was conducted, there were only a few studies that explored how to best make use of such PLMs for TCC. Moreover, previous studies have used different PLMs and different TCC datasets and tasks, making their results difficult to compare. Further, existing widely-used PLMs are trained on formal languages from books and news rather than colloquial online speech where toxic comment can often be found. Therefore, it is unclear to what extent these existing PLMs offer real value for TCC tasks. Our research adds to the current findings

and gives a guideline for future research in this area.

We highlight two directions of transferring PLMs on TCC tasks that are to be explored in this study. The first is the impact of downstream neural-network architectures on the performance of TCC tasks. The second is the benefit of continued fine-tuning PLMs for TCC tasks. What is more, we would like to understand those two strategies cross different language models rather than a single one. In this chapter, we present a study that focuses on these issues above in order to better understand how to transfer PLMs to TCC tasks. We answer the following research questions:

- First, how do different PLMs perform on different downstream TCC tasks, especially multi-class and multi-label classification tasks, which are considered harder than binary tasks?
- Second, how do different downstream neural network architectures impact on the performance on TCC?
- Third, can continued pre-training in domain improve the performance on downstream TCC tasks and if so, can we further improve the effectiveness of the training process while maintaining the performance?

Our results first confirm that using a simple linear neural network as the downstream architecture atop of PLMs for TCC tasks works better than using sophisticated architectures such as CNN and Bi-LSTM, which contain more parameters. Secondly, continued pre-training of a PLM is beneficial to the downstream TCC, especially when the dataset is relatively small. The benefits are also noticeable even with low hyper-parameter settings that make the overall model much more effective compared to those previously reported. This has practical impact as it means that such an approach can be more accessible to users with limited computational resources. Thirdly, the performance of different PLMs varies. BERT and RoBERTa generally outperform XLM on a wide range of TCC tasks, while XLM benefits more from continued pre-training of PLMs.

3.2 Related work

As a brief recap of some of the relevant sections in Chapter 2, traditional text classification has used statistical machine learning methods such as Support Vector Machines, Naïve Bayes and Decision Trees (Schmidt and Wiegand, 2017).

Since 2010 research has shifted towards deep neural networks (DNN)-based models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Bi-directional Long Short-Term Memory network (Bi-LSTM) and hybrid neural networks, which combine different DNN configurations (Del Vigna et al., 2017; Schmidt and Wiegand, 2017; Zhang et al., 2018c). Since the introduction of transformer-based architectures, the use of PLMs in downstream text classification has become the mainstream (Devlin et al., 2018). The basic process is to add task-specific layers for the downstream task atop of the PLM and then train the new model, where only the task-specific layers (i.e., downstream architecture) are trained from scratch (Devlin et al., 2018; Lample and Conneau, 2019; Liu et al., 2019c). Commonly-used PLMs include BERT, RoBERTa, XLM, etc. (Devlin et al., 2018; Lample and Conneau, 2019; Liu et al., 2019c). These models are pre-trained on extraordinarily large corpora, such as those containing over 3 billion words for BERT (Devlin et al., 2018).

Typically, the output layers (the last/bottom few layers following hidden layers) in a PLM are replaced with task-specific downstream architectures and then the new model will be trained in a supervised fashion on the target text classification task. Two strategies are widely used for improving PLMs performance on downstream text classification tasks: the design of downstream neural network architecture and the continued, in-domain pre-training of PLMs (i.e., the "pre-training" phase of the transfer learning in Section 2.4.2 before) (Chronopoulou et al., 2019; Beltagy et al., 2019; Lee et al., 2020; Gururangan et al., 2020).

Downstream network architecture A basic downstream network architecture for classification tasks is a linear transformation layer (Devlin et al., 2018). For example, Munikar et al. (2019) and Mozafari et al. (2019) use a basic downstream architecture on BERT for sentiment classification and hate speech detection. Chronopoulou et al. (2019) extend the basic downstream architecture by adding an extra LSTM layer with self-attention between the linear classifier layer and the PLM. Beltagy et al. (2019) have added two layers of Bi-LSTM on top of BERT. All these different networks, e.g., LSTM with attention, CNN, are added on top of the PLM and are then fed into a classifier layer. There are many other studies using a method of similar fashion. We do not cover them all in detail but point interested readers to the work of Tang et al. (2019); Gao et al. (2019). However, current studies (as at the point of this study) on downstream neural network architectures are mainly based on BERT, and no study has looked at the more recent PLMs such as RoBERTa and XLM.

Continued pre-training of language models Some research explores con-

tinued pre-training of the PLM using a large in-domain corpus (to be referred to as the "fine-tuned" PLM. However, note that this is different from the "fining tuning" strategy in the "adaptation" phase of transferring an LM, as described in Section 2.4.3) and then transferring the newly fine-tuned PLM to the target classification tasks. As a result, training of a model is initialised with weights from a further fine-tuned PLM. This approach is also referred as "continued pre-training in domain" (Gururangan et al., 2020).

SciBERT (Beltagy et al., 2019) and BioBERT (Lee et al., 2020) are two examples of "continued pre-training in domain", which both fine-tune BERT using large domain corpora. Although the continued pre-training in domain by SciBERT and BioBERT only consume unlabelled raw text, they require significant computing resources and time (Gururangan et al., 2020). For example, SciBERT uses a single TPU v3 with 8 cores for one week and BioBERT uses 8 NVIDIA V100 (32GB) GPUs for over 10 days to train BioBERT v1.0 and nearly 23 days to train BioBERT v1.1 (Beltagy et al., 2019; Lee et al., 2020).

To address this, Gururangan et al. (2020) propose task-adaptive pre-training (TAPT) that uses unlabelled training data from downstream target task to continued pre-train RoBERTa. To be more specific, in the stage of continued pre-training of RoBERTa, the label for each data instance in the target task's training dataset is removed. The label-removed training data is taken as the unsupervised learning corpus to further pre-train an existing RoBERTa model. After the continued pre-training, the RoBERTa is trained in a supervised learning fashion on the target labelled training data. This method should be taken differently to the continued pre-training used by SciBERT and BioBERT. Because the continued pre-training and target task fine-tuning of TAPT use the same training corpus, except that in the former case, their labels are removed. On the other hand, PLMs such as SciBERT, BioBERT and ClinicalBERT, use different training data for the continued pre-training and are then transferred to downstream tasks. In comparison, TAPT significantly reduced the data size for continued pre-training. However, in their proposed method, TAPT still requires significant computing resources, such as the computing requirements for the high settings of batch size and the epoch (Kennedy et al., 2020).

Although TAPT is essentially a method of continued pre-training PLMs in domain, in the remainder of this paper, we use "continued pre-training/fine-tuning in domain" to refer to continued pre-training the PLM using a large unlabelled in-domain corpus that is not the same as the training data from the target task. TAPT, on the other hand, will be used to refer to the more specific idea of con-

tinued pre-training in domain using the training data from the target task with their labels removed (i.e., unlabelled target task training data)

3.3 Methodology

We focus on PLM-based methods for TCC and investigate two strategies of transferring PLMs to TCC tasks. First, we analyse the impact of the complexity of the downstream network architectures; Second, we study the impact of continued pre-training PLMs following the TAPT method but using much lower hyperparameters to further reduce the computational requirements. We refer to this as 'TAPT-light' (Gururangan et al., 2020). Third, we compare the performance difference between PLMs.

In short, each model takes a PLM and adds a downstream network architecture on top of it and then all parameters are jointly trained on a given supervised TCC task. The differences between models are based on: the PLM used (Section 3.3.1), the downstream architecture (Section 3.3.2), the continued fine-tuning method of the PLM (whether it is fine-tuned or what is the number of epochs for the fine-tuning, Section 3.3.3). Additionally, we evaluate our strategies using a comprehensive set of TCC datasets, including ten different classes and both multi-class and multi-label classification tasks. Details of these will be given in Section 3.4.1.

3.3.1 Pre-trained language models

To investigate the performance of different PLMs, we choose three of the most cited PLMs to study the generalizability of our strategies applied to TCC tasks. These are BERT, RoBERTa and XLM, representing the current state-of-the-art in a wide range of tasks (Devlin et al., 2018; Lample and Conneau, 2019; Liu et al., 2019c). Another reason we select RoBERTa and XLM is that, as mentioned in Section 3.2, very few studies have used them on TCC, but they have been shown to achieve good performance on other related classification tasks when compared to BERT. Noted that, by the time this study was conducted, most studies of PLMs on TCC tasks only explored BERT. We were the first study on TCC that included two more non-BERT models.

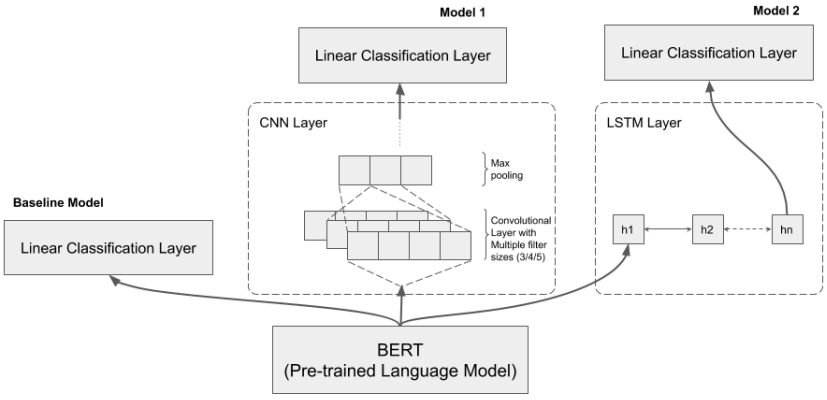

Figure 3.1: *An example of utilizing BERT with different downstream architectures atop.*

3.3.2 Downstream network architectures

In terms of the downstream layers, we compare three architectures: linear classification layer, CNN + linear classification layer, Bi-LSTM + linear classification layer (Devlin et al., 2018; Mozafari et al., 2019). We choose these for the following reasons. First, Bi-LSTM and CNN are frequently used in text classification tasks (Kim, 2014; Chiu and Nichols, 2016). Second, the linear classifier is the simplest architecture and allows us to compare with other complex downstream architectures and their effects on transfer learning performance on TCC tasks.

Figure 3.1 illustrates the three BERT models with different downstream neural network architectures, as an example of a pre-trained model set. RoBERTa and XLM follow the same structure, and each one is tested on three different downstream neural network architectures.

Linear classification layer In the linear layer, the output of the first token (i.e., the output of [CLS]) from the final hidden state of the LM is used in the final classification. The output of a default dimension of given LM is fed to a linear layer and transformed to a dimension equal to the number of labels. A dropout of 0.1 is applied before the linear transformation (Devlin et al., 2018).

CNN + linear classification layer The second architecture is built on the first one but with convolutional layers inserted between the linear layer and the LM. The convolutional layers and architectures have been discussed in the previous Section 2.3.2, which are one of the representative DNNs used in TCC tasks. In the context of our PLM model, the outputs of each layer of the LM (excluding the language modelling head) are fed to a 3-layers convolutional network. This CNN

configuration is the same as that proposed by Kim (2014), which includes three filter window sizes of 3, 4 and 5, respectively. Each size has 100 feature maps. The outputs from each convolutional layer are passed to a max-pooling layer and then concatenated together before feeding to the final linear classification layer, which is the same as mentioned above.

Bi-LSTM + linear classification layer This architecture is similar to the CNN-based architecture, but the three CNN layers are replaced with a bi-directional LSTM layer (Graves and Schmidhuber, 2005). LSTM has been introduced in the previous Section 2.3.2, and it is also a representation DNN used in related TCC studies. In short, the outputs of each layer of the LM are fed to a Bi-LSTM neural network and the final two hidden states of the output are then concatenated before feeding them to the final linear classification layer.

3.3.3 Continued pre-training in domain: TAPT-light

In this part, we explore the performance of continued pre-training in domain using minimum computing resources on TCC tasks. We build our methods on the work of TAPT and reduce the hyper-parameter settings of TAPT to further minimise the computational resources required, which we refer to as TAPT-light (Gururangan et al., 2020). To be more specific, the batch size is reduced to 16, compared to 2048 in the original TAPT. Second, we experiment with different epoch values respectively: 1, 5, 10, 20, 50, 100, compared to 100 in the original TAPT. We focus on batch size and epoch values instead of other hyper-parameter settings, e.g. learning rate, as these two are the main ones deciding the needs of computation resources.

In summary, for each TAPT-light model, the training process is divided into two stages. In the first stage, as highlighted in red in Figure 3.2, the PLM is fine-tuned on unlabelled training corpus that is created from the downstream task (Section 3.4.1). Models are trained for different epochs, respectively, for accessing the impact of epochs later. In the second stage, as highlighted in blue in Figure 3.2, the output layer of the LM is replaced with a linear classification layer and then the new classification model is trained in a supervised fashion on a TCC task, where all parameters are trained jointly. Only the linear classification layers are trained from scratch.

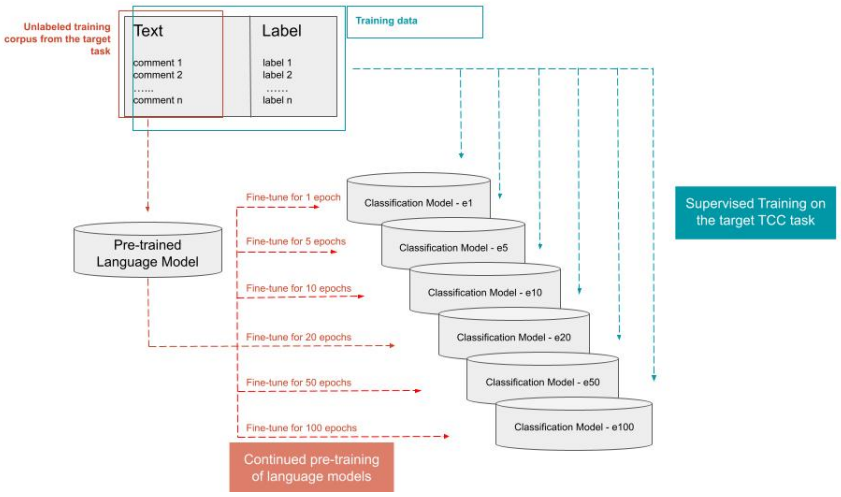

Figure 3.2: *The training process of a TAPT-light model.* Models illustration for the first part of experiment.

Dataset	Data Source	Data Numbers	Labels	Classification Tasks
Kumar (Kumar et al., 2018)	Facebook	14,998	non-aggressive (42%), overtly aggressive (35%), covertly aggressive (23%)	multi-class
Twitter 18k (Waseem, 2016b)	Twitter	18,625	racism (11%), sexism (20%), both (69%), neither	multi-class
Twitter 50k(Founta et al., 2018)	Twitter	50,425	abusive (8%), hateful (3%), normal (73%), spam (16%)	multi-class
Wiki (ConversationAI, 2017)	Wikipedia	159,571	toxic (10%), severe toxic (1%), obscene (5%), threat (0.3%), insult (5%), identity hate (1%)	multi-label

Table 3.1: *Summary of the four toxic comment classification tasks.*

3.4 Experiments

3.4.1 The task and datasets

Four TCC tasks are selected to assess the two strategies of transferring PLMs. The first dataset is collected from Facebook (denoted as *Kumar*) (Kumar et al., 2018). The second and third datasets are both collected from Twitter, denoted as *Twitter 18k* (Waseem and Hovy, 2016) and *Twitter 50k* (Founta et al., 2018). The fourth dataset is collected from Wikipedia Talk page and annotated in a multi-label classification approach, denoted as *Wiki* (ConversationAI, 2017). We aim to cover different social media platforms, dataset sizes and classification types when selecting TCC tasks. The four selected datasets contain between 15,000 and 159,571 comments [1] and cover three different social media platforms with different classification types (multi-class and multi-label). Table 4.1 lists the details of the four datasets. The last three datasets are re-used and converted into binary format in our other studies in Chapter 4 and Chapter 5, which are introduced again.

[1]The original version of Dataset Twitter 50k (Founta et al., 2018) included more than 80,000 tweets with tweets IDs published. We have successfully retrieved 50,425 valid tweets. The remaining missing tweets failed to be retrieved due to their deletion.

3.4.2 Baseline model

We create a baseline model for each PLM with the simplest components possible. Therefore, each baseline model takes a PLM without continued pre-training, and a linear classifier architecture is added on top of it, as shown by "Baseline Model" noted in Figure 3.1.

We select the version for each PLM that is of a parameter number ranging from 110 millions to 144 millions. This way, the three models are of a similar size. Plus, these sizes are smaller than most of their peer versions so that their training is faster. We use the following versions of BERT, RoBERTa and XLM: "bert-base-cased" (12 layers, 768 hidden dimensions, 12 heads, 110 million parameters); "roberta-base" (12 layers, 768 hidden dimensions, 12 heads, 125 million parameters); "xlm-mlm-enfr-1024" (6 layers, 1024 hidden dimensions, 8 heads, 144 million parameters).

3.4.3 Implementation

Pre-processing We constrain the maximum length of each input instance (i.e., a piece of comment) to be 100 tokens, given the available computation resources and the standard practice found in literature (Zhang et al., 2018c). In this way, we truncate the longer comment and pad the shorter messages with zero values.

Hyperparameters For all supervised learning in this work, i.e., the TCC tasks, we train the model for 3 epochs using a batch size of 16, due to the computational resources and empirical experiments results. For all unsupervised learning, i.e., the first stage of TAPT-light, we train the model for different epochs as stated in Section 3.3.3, using a batch size of 16. All training processes use Adam optimizer and a learning rate of 5e-5. The learning rate of 5e-5 (Adam) is from fine-tuning between 5e-5, 3e-5, 2e-5 as these settings are recommended as "possible values to work well" in downstream task-specific training by (Devlin et al., 2018).

Hardware and Implementation We use a single Tesla V100-PCIE-32GB GPU for all experiments. Compared with continued pre-training PLM in domain described in works by Beltagy et al. (2019) (a single TPU v3 with 8 cores) and Gururangan et al. (2020) (Google Cloud v3-8 TPU), our hardware requirements are less powerful. This makes the reproduction of our results easier. Our implementation uses the HuggingFace transformers 3.0.0 library (Wolf et al., 2019) and PyTorch 1.5.1. All multi-class classification tasks use Softmax as the final activation function and the multi-label classification tasks use Sigmoid instead.

Model	**Kumar** (15k)		**Twitter 18k** (18.6k)		**Twitter 50k** (50.4k)		**Wiki** (159.6k)	
	micro F_1	**macro** F_1	**micro** F_1	**macro** F_1	**micro** F_1	**macro** F_1	**micro** F_1	**macro** F_1
BERT	**0.5904**	**0.5805**	**0.8738**	**0.6196**	**0.7957**	**0.6071**	**0.7816**	**0.6372**
BERT-CNN	0.5796	0.5665	0.8480	0.5971	0.7888	0.5929	0.7388	0.4800
BERT-BiLSTM	0.5750	0.5611	0.8469	0.5921	0.7929	0.5959	0.7504	0.5945
RoBERTa	**0.5667**	**0.5441**	**0.8673**	**0.6159**	**0.8025**	**0.6315**	**0.7822**	**0.6510**
RoBERTa-CNN	0.5313	0.4819	0.8491	0.5902	0.7892	0.5367	0.7357	0.4658
RoBERTa-BiLSTM	0.5450	0.5013	0.8426	0.5828	0.7769	0.5563	0.7405	0.5431
XLM	**0.5646**	**0.5503**	**0.8464**	**0.5957**	0.7989	**0.6149**	**0.7594**	**0.5122**
XLM-CNN	0.5525	0.5345	0.8362	0.5860	**0.8023**	0.6015	0.7233	0.4642
XLM-BiLSTM	0.5600	0.5409	0.8346	0.5794	0.7987	0.5751	0.7519	0.4690

Table 3.2: *Comparing F_1 (micro and macro) obtained by transferring different language models (BERT, RoBERTa, XLM) to different downstream network architectures. The best performance for each PLM on each task is boldfaced. The dataset sizes are in the brackets after the datasets and "k" means a thousand or thousands.*

3.4.4 Results: impact of downstream network architectures

Table 3.2 shows the results comparing the different downstream network architectures. "BERT", "RoBERTa" and "XLM" represent the baseline model (as shown by "Baseline Model" in Figure 3.1) for each PLM. As an example, "BERT-CNN" represents the pre-trained BERT PLM with a CNN architecture on top as the downstream network architecture.

We first compare macro F_1 of each model as it highlights how well a model handles minority classes on a unbalance dataset, compared to micro F_1 (Zhang and Luo, 2019). Most of the TCC tasks typically handle highly unbalanced data, and as a result of that, model performance on minority classes (i.e., toxic comments) is often overshadowed by the majority classes when only looking at micro F_1 (Zhang and Luo, 2019). As shown in Table 3.2, the macro F_1 results indicate that the baseline models consistently perform much better than their "complex" counterparts where CNN or Bi-LSTM is used as the downstream network architecture.

In terms of micro F_1, the baseline models obtained the highest F_1 across all datasets, with the exception of XLM on the Twitter 50k dataset. The biggest difference compared to the CNN and Bi-LSTM downstream network architectures were noticed on the smaller datasets, i.e., Kumar and Twitter 18k, and the multi-label classification task (Wiki). For example, the micro F_1 of RoBERTa-CNN is reduced by 0.0465 and 0.0354, compared to the baseline model on the multi-label classification task (Wiki) and the smallest dataset (Kumar) respectively, and reduced by 0.0133 on Twitter 50k, whose dataset is multi-class and three times

Model	**Kumar** (15k)		**Twitter 18k** (18.6k)		**Twitter 50k** (50.4k)		**Wiki** (159.6k)	
	micro F_1	**macro** F_1	**micro** F_1	**macro** F_1	**micro** F_1	**macro** F_1	**micro** F_1	**macro** F_1
BERT	0.5904	0.5805	0.8738	0.6196	0.7957	0.6071	**0.7816**	0.6372
BERT-e1	0.6050	0.5938	0.8737	0.6196	**0.7981**	0.6096	*0.7422*	*0.5462*
BERT-e5	0.6004	0.5899	0.8781	0.6198	0.7965	**0.6185**	*0.7475*	*0.5648*
BERT-e10	0.6029	0.5894	**0.8824**	0.7538	0.7957	0.6076	*0.7804*	**0.6539**
BERT-e20	0.6058	0.5926	0.8743	0.8176	0.7959	0.6079	*0.7404*	*0.6206*
BERT-e50	**0.6092**	**0.5972**	*0.8727*	**0.8190**	*0.7914*	*0.6025*	*0.7426*	*0.5581*
BERT-e100	0.6046	0.5898	0.8765	0.7479	*0.7906*	*0.6046*	*0.7395*	*0.5196*
RoBERTa	0.5667	0.5441	0.8673	0.6159	0.8025	0.6315	**0.7822**	**0.6510**
RoBERTa-e1	0.6075	0.5924	0.8690	0.6160	**0.8106**	0.6382	*0.7393*	*0.5067*
RoBERTa-e5	0.6075	0.5961	**0.8808**	0.6277	*0.8019*	**0.6391**	*0.7606*	*0.5650*
RoBERTa-e10	0.6146	**0.6051**	0.8711	0.6174	0.8060	0.6368	*0.7530*	*0.5606*
RoBERTa-e20	0.6063	0.5922	0.8711	*0.6134*	*0.8011*	*0.6275*	*0.7736*	*0.5875*
RoBERTa-e50	0.6146	0.6045	0.8759	**0.8205**	*0.7997*	*0.6163*	*0.7711*	*0.6503*
RoBERTa-e100	**0.6167**	0.6043	0.8743	0.7439	0.8027	*0.6242*	*0.7683*	*0.6209*
XLM	0.5646	0.5503	0.8464	0.5957	0.7989	0.6149	0.7594	0.5122
XLM-e1	0.5754	0.5566	0.8593	0.6063	**0.8044**	**0.6248**	*0.7548*	0.6247
XLM-e5	0.5883	0.5762	0.8593	0.6057	*0.7987*	0.6184	0.7600	0.5837
XLM-e10	0.6000	0.5896	0.8636	0.7364	0.8003	0.6170	0.7661	0.5952
XLM-e20	0.5988	0.5891	**0.8695**	**0.7826**	*0.7987*	0.6184	0.7712	0.6006
XLM-e50	0.6158	0.6091	0.8690	0.7384	0.7999	0.6178	0.7707	**0.6427**
XLM-e100	**0.6196**	**0.6097**	0.8727	0.7883	*0.7985*	0.6155	**0.7748**	0.6233

Table 3.3: *Comparing F_1 (micro and macro) of the baseline models and their continued domain fine-tuning language model counterparts. "BERT-e1" refers to fine-tuning the BERT language model for 1 epoch and then transfer it to the target TCC tasks. The best performance for each PLM on each task is boldfaced. The model whose result is worse than its baseline model is italicized.*

bigger than Kumar. Similar patterns are observed on results of BERT groups and XLM groups. Comparing CNN and Bi-LSTM as the downstream architecture for the PLM, there is no consistent pattern indicating which one is better.

The results above suggest that transferring a PLM to complex downstream network architectures does not offer any benefits over a simple linear architecture in the context of TCC tasks. One possible reason for this is that the complex DNN might dilute or cannot interpret well the general representations that PLMs learned from pre-training on extremely large corpora.

3.4.5 Results: impact of continued pre-training in domain

In summary, we have tested TAPT-light on four TCC tasks using three public PLMs for six different epoch settings. In total, 72 models have been reported in Table 3.3.

Firstly, as shown in Table 3.3, TAPT-light often leads to further performance gain compared to the baseline models, with exceptions noted on the Twitter 50k dataset and Wiki dataset when using BERT and RoBERTa. Particularly on the smaller datasets, Kumar and Twitter 18k, the performance improvements gained

from TAPT-lights are noticeable. For example, the macro F_1 on Twitter 18k has increased by 0.1994, 0.2046 and 0.1427 for BERT, RoBERTa and XLM respectively, when further pre-training the PLM for 50 epochs using the unlabelled downstream data. The higher increase of macro F_1 than micro F_1 also suggests that continued pre-training particularly benefits small classes in the datasets, and these are often classes of the toxic comment rather than the non-toxic comment. While for Wiki, the biggest dataset among the four, the baseline models of BERT and RoBERTa achieved a strong performance and TAPT-light demonstrates a detrimental impact on its classification performance. The only exception is XLM. The three continued PLMs give no noticeable advantages or disadvantages over the Twitter 50k task.

Another finding is that there is no consistent pattern in terms of which epoch setting is the best across the four tasks. However, for multi-class classification tasks (i.e., Kumar, Twitter 18k and Twitter 50k), models fine-tuned for 10 or 20 epochs achieve the highest F_1 scores or comparable results to their counterparts that use a higher epoch value. For the multi-label classification task (i.e., Wiki), as mentioned above, continued fine-tuning in domain does not show noticeable benefits on this task. The dataset of this task is at least three times bigger than the other three.

To summarise, our results suggest that TAPT-light is beneficial to TCC tasks on relatively small datasets. This may be because a large dataset could have already supplied adequate information for the model to learn; thus, features from PLM offer less value to the learning. Additionally, for TCC tasks, more training epochs do not necessarily lead to performance benefits. Also, under these circumstances, a small batch size, e.g., 16, is workable. One possible explanation is that a large number of epochs could have led to catastrophic forgetting that useful features gained from the massive PLM pre-training could be erased during the later TAPT-light training and thus lead to a decrease of model performance (Ruder, 2019). Although we did not directly compare against TAPT, we believe our findings are still encouraging for future researchers since a identical reproduction of TAPT is very resource demanding that is not accessible to most people in realistic situations. Our results suggest that even with very low hyperparameter settings, continued fine-tuning in-domain still contributes to TCC task especially when the training dataset of the TCC is relatively small and unbalanced.

3.4.6 Results: impact of different PLMs

As shown in Table 3.2, BERT generally outperforms RoBERTa and XLM when an identical downstream architecture is applied. This pattern is clearer when comparing their macro F_1 and particularly on small datasets like Kumar. For example, BERT-CNN gains a macro F_1 of 0.5665 on Kumar, noticeably higher than RoBERTa (0.4819) and XLM (0.5345).

With TAPT-light (results in Table 3.3), RoBERTa generally outperforms BERT and XLM, especially when comparing their best TAPT-light models and baseline models on each TCC tasks with regard to macro F_1. Moreover, TAPT-light benefits XLM more than RoBERTa and BERT, particularly on TCC tasks with relatively small datasets. This can be observed as that all TAPT-light XLM models outperform its XLM baseline in terms of macro F_1.

One possible reason behind the above patterns is that XLM handles long sentences or documents (4000 tokens composed of sentences) during its pre-training, while TCC usually tackles short sentences and texts (Lample and Conneau, 2019). TAPT-light uses in-domain data to fine-tune PLMs, and therefore, could have helped XLM to 'learn' short texts. On the other hand, BERT and RoBERTa were pre-trained on corpora whose maximum sequence length is 512 tokens that is similar to many TCC datasets (Devlin et al., 2018; Liu et al., 2019c). This could have explained why they generally outperform XLM and that they benefit less from TAPT-light.

3.4.7 Limitations

The current experimental results do not demonstrate a consistent pattern over 4 datasets, which might lead to a weak takeaway for future researchers in TCC. However, we argue that our work still contributes to the current understanding of using PLM on TCC tasks and bridges the gap where there was a lack of comprehensive comparison of PLM models on a wide range of TCC tasks. We believe our work also encourages future TCC studies to further consider the selection of PLMs and parameters settings with an attention of the dataset size. Second, being limited to computational resources, this study lacks a set of experiments with identical parameter settings to TAPT, which might give further information on how TAPT-light compares to the original TAPT on TCC tasks. This is a common issue of reproducibility for much research that require expensive hardware (Belz et al., 2021). However, this is also our motivation to propose TAPT-light as it require much less computational resources compared to the original TAPT. Again,

due to the unavailability of the original datasets[2] used in TAPT, we were unable to directly compare our TAPT-light against TAPT on identical datasets. Another unexplored factor is that each PLM and sophisticated architectures such as CNN and Bi-LSTM could have their own favourable hyper-parameter settings. However, those hyper-parameter settings are unified in our study according to the common settings in previous studies but not further refined. Also, to what extent those hyper-parameter settings impact on the performance is unknown.Similarly, different versions of PLM (this study only tested the base version for BERT, RoBERTa and XLM) might also tell us a different story. How different versions of PLM impact the results is worth exploring in the future. Last, by the completion time of this study, more PLMs are proposed in the latest research, such as TweetBERT, DEBERTA (He et al., 2020; Qudar and Mago, 2020). It will add up to the current understanding of using PLMs on TCC tasks if our studies can be generalized to these newer PLMs.

3.5 Conclusion

This chapter studied transferring PLMs to TCC tasks. We focused on the impact of different PLMs, downstream network architectures, and using the dataset of the downstream task to continue fine-tuning the PLM before transferring them to downstream TCC tasks, as well as the impact of lowering some of their training hyper-parameters. Our comprehensive experiments have provided evidence-based answers to the research questions proposed in the beginning of this Chapter.

First, regarding the performance of different PLMs on different downstream TCC tasks, our results suggest that BERT and RoBERTa generally perform better than XLM. Second, regarding the impact of different downstream architectures, we have shown that using a simple downstream network architecture is a better choice over a complex one, such as CNN and Bi-LSTM. Third, regarding the effectiveness of TAPT even using less computational resources (i.e., TAPT-light), our results of TAPT-light show that, all other variables being equal, a low epoch setting will suffice to obtain the best achievable results of a model in most cases. This potentially makes the TAPT-based methods more accessible and easier to train.

Our future work will explore several directions. For example, we will explore these studies in a different aspect of downstream tasks to see if our findings gen-

[2]Here we refer to the two datasets in the REVIEW domain that is a relatively similar domain to TCC.

eralise well on them; we will develop and test systematic methods of unfreezing and freezing certain layers' parameters in the PLM.

Chapter 4

The identity term bias in TCC

In the last chapter, we introduced SOTA DNNs, PLMs and investigated different PLM transferring strategies for TCC tasks. While such methods benefit from knowledge transferred from pre-training and avoid ad-hoc feature engineering, there is less control over what features are learned by the model. A downside of this is that the model can inherit unintended bias from data that are used for pre-training. Several recent studies find that DNN-based models such as PLMs can be biased toward identity terms on TCC tasks (Park et al., 2018; Dixon et al., 2018; Kennedy et al., 2020). Therefore, this study aims to investigate this bias on a wide range of TCC datasets to gain a deeper understanding of such bias. We focus on the SOTA PLM, BERT, which is also the most frequently used NLP model since 2017 (Vaswani et al., 2017).

The remainder of this chapter is structured as follows. Section 4.1 briefly explains the concepts and our study. Section 4.2 reviews different bias types studied by previous work on TCC and how related work approach bias. Section 4.3 explains the tasks, datasets and models we use to investigate the identity term bias and defines the concept of "subjectivity". Section 4.4 presents our experiment details, including model explanations using rationales, the qualitative and quantitative analysis of model prediction results (Lei et al., 2016; Jacovi and Goldberg, 2020). Section 4.5 concludes this chapter and discusses future work.

4.1 Introduction

DNN-based toxic comment classification models are often found suffering from unintended bias such as bias towards gender, race and nationalities. A few previous TCC work have defined the issue as identity term bias (Park et al., 2018; Dixon

et al., 2018; Kennedy et al., 2020). Identity terms are terms characterising a specific group of people such as "Muslim" and "black". When a model has identity term bias, it tends to predict a comment as toxic if it contains identity terms. The identity term bias is commonly reflected in false positive predictions, i.e. non-toxic comments with identity terms. Such bias not only affects model performance but also discourages an open and friendly discussion if found common in real applications. Limited studies have been done to combat such bias which leads to several research gaps that are worth addressing.

First, previous studies on the identity term bias are limited to using one or two datasets that may not be representative of the problem. An investigation of a wide range of TCC datasets will contribute to a deeper understanding of the identity term bias and also produce generalizable findings. Second, although previous studies have looked at the identity term bias, they mainly investigate surface patterns, e.g., counts of comments with identity terms or what identity terms are associated with false positive predictions. Evidence and analysis beyond surface patterns over individual examples are needed. An investigation into the decision-making rationales of the model can help understand the impact of the identity term bias for a model. Third, to the best of our knowledge, no previous studies of bias in TCC models have looked at the subjectivity level of comments. A comment with a low subjectivity level expresses more factual information and less personal feelings and opinions; while a comment with a high subjectivity level contains more personal opinions but less factual information. Instead of focusing on word-level features, we are interested in how the subjectivity level of the comment interacts with the identity term bias.

Most toxic comments such as hate speech, aggressive language and abusive language, tend to express hate or encourage violence towards a person or group based on certain characteristics such as race, religion, sex, or sexual orientation (Gitari et al., 2015). Such expressions are intuitively more a reflection of personal feelings rather than fact-quoting. Therefore, we propose a hypothesis that when a comment contains an identity term and has a low level of subjectivity, it tends to be non-toxic. For example, the comment of

"that land has been the home of jews for 3000 years."

should not be taken as a toxic comment as it attempts to express a factual information regardless of whether it is true. On the other hand, when a comment contains an identity term and has a high level of subjectivity, it tends to be toxic.

For example, "this jew idiot is sp ##ew ##ing nothing but propaganda." should be taken as a toxic comment as it expresses a subjective and offensive opinion. Here, we are interested in examining the subjectivity level rather than taking it as a binary detection problem as most related work have done. On top of it, we propose different methods to examine the subjectivity level.

To address the limitations above, we conduct an investigation of the identity term bias over a wide range of TCC tasks with both qualitative and quantitative methods. Particularly, we investigate the interaction between the subjectivity levels and the identity term bias, which is based on an explainable intuition as mentioned above. Therefore, the research questions of this chapter are:

- First, to what extent is the identity term bias found in SOTA methods for TCC tasks?
- Second, how can we identify the evidence of the identity term bias in a TCC model?
- Third, what is the relationship between the subjectivity level of a comment and the identity term bias of a TCC model?

Our investigation first confirms that the identity term bias is a common issue found in TCC models. The model explanations we have extracted provide further details of how the identity term bias impacts the model behaviour. Second, our qualitative analysis and quantitative analysis demonstrates the relationship between the identity term bias and the subjectivity level of comments. Among comments with identity terms, the false positive predictions, i.e., non-toxic comments, tend to have a low level of subjectivity. While, the true positive predictions, i.e., toxic comments, tend to have a high level of subjectivity. However, this pattern cannot be observed among comments without identity terms.

4.2 Related work

Machine learning models are designed to identify and use biased patterns in data to help the prediction tasks. For example, a model trained to identify toxic comments is intended to be biased towards features of toxic comments such that toxic comments receive higher scores than those which are not toxic (Dixon et al., 2018). Nonetheless, the model is not designed to discriminate people based on the groups, classes, or other categories to which they belong to or are perceived to belong to,

such as gender, religion and race. If it does, we refer to this type of "discrimination" learned by the model as unintended bias (Dixon et al., 2018).

A few previous work have discussed the harmful impacts of applying such unfair models for real-world tasks, while leaving the solutions open (Hovy and Spruit, 2016; Blodgett and O'Connor, 2017; Tatman, 2017; Dixon et al., 2018). For example, Obermeyer et al. (2019) have found that an algorithm used in US hospitals for allocating health care systematically discriminates against black people, since it is less likely to refer black people than white people who are equally sick to programs that aim to improve care for patients with complex medical needs. In the field of computer vision, some facial recognition algorithms labels black people as "gorillas" (Howard and Borenstein, 2019). In the field of audio processing, it is found that voice-dictation systems recognize a voice from a male more accurately than that from a female (Rodger and Pendharkar, 2004; Garrido-Muñoz et al., 2021). What is more, Tolan et al. (2019) have identified that some risk assessment systems are likely to predict people of certain races to be more likely to commit a crime. In the field of NLP, Hovy and Spruit (2016) have found that restricted language, like class specific language or scientific jargon, can hinder the expression of outsiders' voices from certain practices. Blodgett and O'Connor (2017); Dixon et al. (2018) show that the use of dialect of African-American English can make a comment more likely to be predicted as toxic regardless of its content. This bias towards African-American English may cause relationships between different ethic groups deteriorate.

As per the definition used by Hardt et al. (2016) and Dixon et al. (2018), *a model contains unintended bias if it performs better for some demographic groups than others.* A few recent work separate different unintended biases based on demographic features, such as gender bias, racial bias, and dialectal bias (Tan and Celis, 2019; Sap et al., 2019; Davidson et al., 2019; Zhou et al., 2021; Park et al., 2018; Kennedy et al., 2020; Bolukbasi et al., 2016; Zhang et al., 2020; Vaidya et al., 2020; Halevy et al., 2021). Studies by Davidson et al. (2019); Mozafari et al. (2020); Xu et al. (2021) focus on racial bias against users using African-American English. They find that tweets written in African-American English are predicted as toxic significantly more often than those written in standard American English. While this kind of bias is represented as dialectal bias, Mozafari et al. (2020) and Halevy et al. (2021) both refer to it as racial bias. Bolukbasi et al. (2016) and Zhang et al. (2020) study the gender bias found in the pre-trained Word2Vec word embeddings. Bolukbasi et al. (2016)'s study show that Word2Vec contains female/male gender stereotypes. For example, the words like "receptionist" and

"she" are strongly associated to each other, so are "maestro" and "he".

Park et al. (2018); Dixon et al. (2018); Kennedy et al. (2020) introduce "identity term" bias. "Identity terms" (also known as "group identifiers") are words or terms referring to people with specific demographic characteristics, such as ethnic origin, religion, gender, or sexual orientation. Park et al. (2018); Dixon et al. (2018); Kennedy et al. (2020) point out that TCC models tend to assign too much attention to such identity terms, resulting in incorrect predictions. Such bias towards identity terms often reflects the false positive predictions, known as false positive bias (Dixon et al., 2018; Halevy et al., 2021). For example, Park et al. (2018) give an example in their study that "You are a good woman" is predicted as "sexist". One concept closely-related to "identity terms" is "bias sensitive words", proposed by Badjatiya et al. (2019). They define that *a word w is defined as a bias sensitive word for a classifier if the classifier is unreasonably biased with respect to w to a very high degree.* For example, as discussed in their work, "dirty", "shit", "gotta", "muslims", "she" and "woman" are bias sensitive words (Badjatiya et al., 2019). The main difference between "identity terms" and "bias sensitive words" is that a bias sensitive word is unnecessarily an identity term referring to a group of people. Another point worth mentioning is that racial bias and gender bias cannot be taken as sub-concepts or sub-type of the identity term bias directly as racial bias and gender bias in language do not have to contain identity terms. For example, racial bias can be reflected in the bias against African-American English and gender bias can be reflected in the gender stereotypes regarding professional positions existing in the model.

The identity term bias often leads to false positive predictions in TCC. The scenario is that a TCC model tends to predict a non-toxic comment as toxic when it contains identity terms. Previous TCC studies on identity term bias have analysed which terms are influential identity terms to a TCC model. For example, Kennedy et al. (2020) utilize the Sampling and Occlusion (SOC) algorithm to capture the tokens that account for the model predictions and then manually select identity terms from those tokens. Dixon et al. (2018) link the identity terms bias to the data distribution due to the disproportional usage of identity terms in toxic comments. For example, in the dataset they study, the word "gay" appears in 3% of toxic comments but only 0.5% of comments overall. They believe the disproportional amount of toxic comments with identity terms in the training dataset leads to the model overfitting. To the best of our knowledge, there are few published papers studying the identity term bias in terms of the subjectivity levels of comments.

Previous studies show that toxic comments, such as offensive speech and hate

speech, tend to be expressions of subjective feelings or opinions (Pang and Lee, 2004; Gitari et al., 2015; Burnap and Williams, 2016; Benito et al., 2019). A few TCC studies have utilized subjectivity in classification. For example, Gitari et al. (2015) assume non-subjective sentences are not toxic comments and therefore they filter out non-subjective sentences with a rule-based approach prior to classification. These filtered sentences are considered automatically to contain non-hateful content. The intuition is to make the classification task "easier" by removing non-subjective sentences in advance. The study by Van Hee et al. (2018) use the positive and negative opinion word ratios and the polarity calculated with sentiment lexicons as the "subjectivity lexicon features" of a comment. They find that subjectivity lexicon features prove to be strong features for cyberbullying detection. In short, these studies have implied that the likelihood of a comment being toxic is associated with its subjectivity. However, none of them attempts to quantify the subjectivity level or utilize it in a non-binary fashion to improve the model. Furthermore, none of them has looked into how subjectivity can be used to address the identity term bias.

Although a few TCC studies explore utilizing subjectivity in TCC tasks, there is a lack of consensus on how subjectivity should be defined. For example, Gitari et al. (2015) briefly point out that "a subjective sentence expresses some feelings, views, or beliefs." Other studies, such as Lin et al. (2011) and Huo and Iwaihara (2020), directly focus on subjectivity detection but none of them gives a precise definition of subjectivity or an explanation of what kind of comments or texts are supposed to be labelled as "subjective".

In summary, a growing body of TCC research have studied the identity term bias. However, these studies are limited to a small range of datasets in their studies, which make the findings not generalizable enough. Furthermore, current studies discern the identity term bias by analysing the prediction results and the word-level features. Far too little attention has been paid to the evidence of the identity term bias using model rationales. In other words, very litter work has tried to explain the model's behaviour of identity term bias from the perspective of the model decision-making process. Last, no TCC research have investigated the relationship between the subjectivity level and the identity term bias.

4.3 Methodology

We train a SOTA NLP model, BERT, on a wide range of TCC tasks separately to investigate the identity term bias in TCC models. First, we look into the model

rationales on the false positive predictions that contain identity terms. The aim is to seek the evidence of the identity term bias using a model's rationales. Second, we hypothesise that when a comment has identity terms, its subjectivity level can be associated with its toxicity. Intuitively, a comment with an identity term tends to be toxic if it has a high level of subjectivity, on the contrary, it tends to be non-toxic if it has a low level of subjectivity. To verify this hypothesis, we conduct qualitative and quantitative analysis of model prediction results, considering a ternary relationship among the subjectivity level of a comment, the presence of identity terms and the prediction results.

4.3.1 The task and datasets

We include four datasets in the analysis, with the aim to cover different social media platforms, dataset sizes and text lengths. We reuse 3 datasets from the last chapter, namely Twitter 18k (Waseem, 2016b), Twitter 50k (Founta et al., 2018) and Wiki (ConversationAI, 2017). We replace the smallest dataset Kumar (Kumar et al., 2018) with a hate speech dataset collected from a white supremacist online forum (denoted as *WS*) (de Gibert et al., 2018), which has a similar size to Kumar (WS has 10,703 posts and Kumar has 14,998 posts). The main reason for the replacement is that WS is employed by Kennedy et al. (2020) to study the identity term bias, thus using the same dataset can allow fair comparisons to their method. The WS dataset includes 10,703 posts in total and 1,196 of them are "Toxic", 9,507 are "non-Toxic" [1]. For task Twitter 50k, we remove the data labelled as "spam" to keep the focus of this chapter is on TCC tasks. This has reduced Twitter 50k to 42,314. To distinguish the two versions of this task, we denote Twitter 50k without "spam" as "Twitter 42k".

Since the identity term bias is found in various types of toxic content and also to follow the practice by Kennedy et al. (2020) which studies the identity term bias in the context of binary classification, we group different toxic comments into one group without distinguishing their specific types for task Twitter 18k, Twitter 42k and Wiki. Therefore, the task in this work is a binary toxic comment classification task where the model aims to predict if the comment is toxic or not. Previous study has also followed a similar practice (Beatty, 2020). Table 4.1 as shown below summarises the four datasets used in this chapter.

[1]The original binary labels are "hate" and "no hate".

Dataset	Source	Data Numbers	Original Labels	Toxic Proportion	Avg Text Length
WS (Kumar et al., 2018)	Stormfront	14,998	non-aggressive (42%), overtly aggressive (35%) covertly aggressive (23%)	11.17%	91
Twitter 18k (Waseem, 2016b)	Twitter	18,625	racism (11%), sexism (20%), both (6.9%), neither	31.22%	96
Twitter 42k(Founta et al., 2018)	Twitter	42,314	abusive (9%), hateful (4%), normal (87%)	13.48%	123
Wiki (ConversationAI, 2017)	Wikipedia	159,571	toxic (10%), severe toxic (1%), obscene (5%), threat (0.3%), insult (5%), identity hate (1%)	10.17%	398

Table 4.1: *Summary of the four toxic comment classification tasks. "Toxic Proportion" refers to the proportion of "Toxic" comments after the conversion to binary classification.*

4.3.2 Predictive model: BERT

Influenced by Kennedy et al. (2020), we use BERT as the classifier to demonstrate the problem of identity term bias in TCC tasks. BERT would be a good baseline as it is widely used in various NLP tasks and many pre-trained language models are built upon it (Devlin et al., 2019; Liu et al., 2019c; Lample and Conneau, 2019). We ask readers to refer to Section 2.5.2 for details of BERT.

4.3.3 Model explanations with rationales

To study whether and how the BERT classifier is affected by the identity term bias during its decision-making process, we look into methods of "model explanations" to explain the rationalisation behind the model predictions (Lei et al., 2016; Jacovi and Goldberg, 2020; Chrysostomou and Aletras, 2021). One commonly-used approach to obtain model explanations for DNNs is computing an importance score for each input token to identify which parts of the inputs contributed the most towards the model prediction (Jacovi and Goldberg, 2020; Chrysostomou and Aletras, 2021). Tokens with high importance scores are taken as the evidence or reasons for the model to make such predictions. These tokens are known as rationales (Jacovi and Goldberg, 2020; DeYoung et al., 2020; Chronopoulou et al., 2019). Intuitively, for a classification task, the model will potentially flip the predicted label when taking inputs without these rationales. For example, the rationale for the toxic comment *"feminist is very stupid"* could be *"stupid"* or *" 'feminist' + 'stupid' "*. An ideal model should not predict the comment *"feminist is"*, *"is very stupid"* or *"feminist is very okay"* as toxic when *"feminist"* or *"stupid"* is removed from the input. Previous studies have used rationales to investigate the data or to gain insights from the data. For example, Jiang and Wilson (2021) extract the rationales from a misinformation dataset and then cluster those extracted rationales to investigate the misinformation types.

Different methods are used to compute the importance scores. One approach is to calculate the gradients of a prediction with respect to the input token (Jain

et al., 2020; Atanasova et al., 2020). The intuition is that, for any tiny change in the input, the gradient indicates how the output would change. The bigger change in the output, the more the model takes the input token into account for the prediction. Within transformers-based models, such as BERT, attentions for the input are also commonly-used to calculate importance scores. Since attention-based models calculate attention distributions over inputs given a token, prior work assume that the attention over one token intuitively reflects the importance the model assigns to tokens and tokens with high attentions potentially play important roles in the model prediction (Wang et al., 2016b; Lee et al., 2017; Vaswani et al., 2017). Further, Jain et al. (2020) believe that the attention-based importance scores reflect the importance not of individual inputs, but rather of unknown interactions between tokens and all input tokens.

The quality of extracted rationales are usually measured by the agreement with human-provided rationales or assessments of faithfulness (DeYoung et al., 2020). Faithfulness here refers to evaluation measures or metrics that do not require human-provided rationales (Lei et al., 2016; DeYoung et al., 2020; Jacovi and Goldberg, 2020). It reflects the degree to which the extracted rationale affects the corresponding prediction (Lei et al., 2016; DeYoung et al., 2020; Jacovi and Goldberg, 2020). For example, sufficiency is a faithfulness metrics that measures the change of the model prediction performance when only using the rationales as the input (DeYoung et al., 2020). Faithful rationales with a high level of sufficiency are expected not to lead to a big drop of model prediction performances. On the other hand, comprehensiveness measures the change of the model prediction performance when masking the rationales in the input (DeYoung et al., 2020). Faithful rationales with a high level of comprehensiveness are expected to see a big drop of model prediction performances when rationales are masked in the input.

Although there have been debates on the faithful performance by attention-based explanations, rationales extracted by straightforward rules on attention weights are demonstrated to be comparable to human-provided rationales (Wiegreffe and Pinter, 2019; Serrano and Smith, 2019; Jain et al., 2020). In this study, our main focus is to investigate the model behaviour regarding the identity term bias rather than to generate faithful rationales. Therefore, we take advantage of the built-in access to both the gradients and the attention weights to computing the importance scores for tokens. For attention-based importance scores, we calculate the normalised attention weights for tokens (Jain et al., 2020). For gradients-based importance scores, we multiply the input with its gradient computed with

respect to the predicted class (Kindermans et al., 2016; Atanasova et al., 2020). For both methods, we take tokens of the top 20% highest importance scores as the rationales for the comment (Chrysostomou and Aletras, 2021). We apply this analysis to false positives predicted by the model on the four datasets to analyse the identity term bias.

4.3.4 Defining "Subjectivity"

To the best of our knowledge, there is no consensus on the definition of subjectivity. In this work, we adopt the definition of "subjectivity level" by TextBlob as mentioned before: the subjectivity level describes the extent to which the comment conveys personal opinion or factual information (Loria, 2018). A comment with a high level of subjectivity indicates that the comment contains more personal opinion and less factual information, vice versa.

The TextBlob[2] library (Loria, 2018) is the tool this work uses to generate subjectivity scores to facilitate our bias analysis. We have also identified another similar tool, SentiWordNet (Sebastiani and Esuli, 2006). Given a text, both tools assign a subjectivity score within the range from 0.0 to 1.0 where 0.0 is very objective and 1.0 is very subjective. No report is found comparing the accuracy of TextBlob and SentiWordNet. We conducted our own analysis and concluded that TextBlob is more accurate at predicting subjectivity over the four TCC datasets.

First, we compare the subjectivity scores given by TextBlob and the ones by SentiWordNet for each comment from the four datasets. As shown in Figure 4.1, SentiWordNet assigns most texts an extreme score of "1" (very subjective). The total percentages of "1" and "0" assigned by TextBlob over the four datasets are 39.76%, 42.44%, 32.86% and 21.85%. However, for SentiWordNet, the percentages are 76.07%, 74.30%, 74.27% and 66.13% respectively. In other words, SentiWordNet has a distribution biased towards "1", while TextBlob is less biased towards the two extremes.

Second, we conduct a manual inspection, validating the output of the two tools on a sample of 80 data instances from the 4 datasets (20 per dataset). We randomly select these 80 comments whose subjective scores by the two tools have a disagreement greater than 0.5. To give an example, the following comment:

"my mothers father is full blonde Irish. my mothers mother is English Irish Scottish . my fathers father is a German with some Scottish and my fathers mother

[2]https://textblob.readthedocs.io/en/dev/#

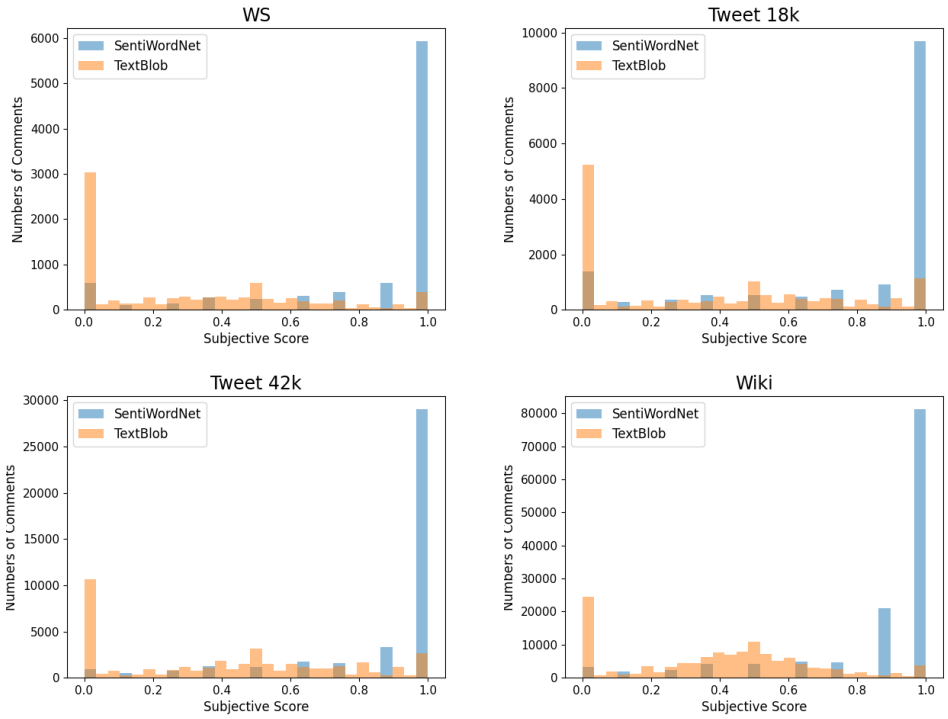

Figure 4.1: *The subjective scores distribution of TextBlob and SentiWordNet over the four datasets. The X-axis is the comment count of corresponding subjectivity scores indicated by the Y-axis. All four plots are based on the training dataset.*

is danish and Norwegian. making me 6ft4 blonde hair blue eyes with a big barrel chest just like my german grandfather .”

is scored differently by TextBlob (*0.0682*) and SentiWordNet (*1*). In this case, we take the score *0.0682* by TextBlob as the more accurate one. Our manual investigation shows that TextBlob is more accurate for 77.5% samples we validated. The full comparison between TextBlob and SentiWordNet over the 80 comments are presented in the appendix.

Based on this qualitative analysis and the score distributions, we believe the subjectivity scores with a more refined granularity by TextBlob will offer the model more information on the subjectivity level of comments. Therefore, we use TextBlob to measure the subjective level.

4.4 Experiments

4.4.1 Implementation

We train the initial base uncased version of BERT model ("bert-base-uncased" from the software package by Wolf et al. (2019), the same one as described in Section 3.3.2 with a linear classification layer atop) on the four TCC tasks separately and investigate their prediction results. For each task, a model runs 10 times independently to give a mean value on each evaluation metrics, following the implementation in Kennedy et al. (2020).

We constrain the maximum length of each input instance to be 128 tokens for WS, Twitter 18k and Twitter 42k, and 400 tokens for Wiki. The maximum length is set up based on the average length of text in each dataset as detailed in Table 4.1, which aims to cover the full text of most data and saves computing resources at the same time.

The hyperparameter settings follow those in Kennedy et al. (2020) to provide a direct and fair comparison to their work. Accordingly, the batch size is set to 32. Adam optimization is implemented with a starting learning rate of 2 x 10-5. The validation is performed every 200 steps and the learning rate is halved every time the validation F_1 decreases. The model stops training after the learning rate halved 5 times. We also re-weight the training loss to handle the imbalance labels as Kennedy et al. (2020) does. Note that several hyperparameters and implementations are different from what we did in Chapter 3 as we aim to compare with the work by Kennedy et al. (2020) in this study. Also, Chapter 3 studied the impact of hyperparameters so different sets of hyperparameters were tested there. But the impact of hyperparameters is not the focus of this study.

We used a single NVIDIA Tesla V100 GPU for all experiments. We build the BERT model with the code provided by Kennedy et al. (2020), such that our implementation uses the same software packages by Wolf et al. (2019) and Paszke et al. (2019). Our rationalisation code is built upon the code by Chrysostomou and Aletras (2021), which also heavily uses the packages by Wolf et al. (2019) and Paszke et al. (2019).

4.4.2 Model explanations with rationales

Before analysing the model predictions on comments with identity terms, we investigate model decisions with respect to the identity term bias. We extract rationales for false positive predictions whose comments contain identity terms There are a

total of 137 such comments across all datasets, and we manually analyse them. Noted that, due to the nature of sensitivity of TCC tasks, we emphasise the importance of handling identity term bias even identity term bias is only reflected on a small part of instances. We find that 125 (91.24%) comments have rationales with identity terms (attention-based or gradients-based or both). Examples are shown in Table 4.2. The high importance scores assigned to the identity terms potentially contribute to the erroneous predictions. Take comment **3** in Table 4.2 as an example, if we simulate the model decision-making process where we pay more attention to the rationales "whites" and "negroes", we might label this comment as toxic. However, if we examine the whole comment, we might change our decision from toxic to non-toxic as this comment is making a metaphor that indicates "whites" and "negroes" are just different rather than one is better the other. In short, our analysis has shown evidences that SOTA TCC classifiers over-attend to identity terms, this could lead to false positive predictions.

4.4.3 The identity term bias and subjectivity: qualitative analysis

Here we study the relation between the identity term bias and the subjectivity level. Errors made by BERT are extracted for our analysis, i.e. false positive predictions and false negative predictions. As mentioned before, we use TextBlob to generate a subjectivity score for those comments (Loria, 2018). We investigate those errors and select several representative examples as shown in Table 4.3. We observe that to correctly predict the toxic comments with identity terms, we need to account for the meaning of the whole sentence and the stance of the speakers. Subjectivity level is possibly one perspective from which this can be captured. For example, in the first pair which mentions "muslims" in Table 4.3, the toxic comment compares "blacks" with "muslims" and asserts that "they (Muslims) are worst than blacks" without any factual information, this indicates a high level of subjectivity. While the non-toxic comment with the term "muslims" attempts to describe an objective fact related to Muslims, although it contains comparison, it attempts to include a specific figure to support the comparison[3]. This observation inspires us that the subjectivity score of the comments can be a helpful indicator when classifying a comment with identity terms.

[3]The "fact" and figure the comment provides are unnecessarily true but it is not the topic we aim to study.

4.4.4 The identity term bias and subjectivity: quantitative analysis

To further analyse the relation between the subjectivity level and the identity term bias at scale, we conduct a quantitative analysis of the subjectivity score of false positive and true positive predictions. We separate those comments with identity terms and those without identity terms to examine the identity term bias. The identity term list is adopted from Kennedy et al. (2020) which includes 25 terms such as "muslim", "jew", and "women", full list is shown below (in Table 4.4.4). The main reason we use this identity term list is that we want to build up our method (in the next chapter) based on the observation in this quantitative analysis and we would like to compare our method to the method by Kennedy et al. (2020). Thus we use the same list as they do to avoid the impact of different identity term lists on the model performance.

muslim jew jews white islam blacks muslims women whites gay black democat islamic allah jewish lesbian transgender race brown woman mexican religion homosexual homosexuality africans

Table 4.4: *25 identity terms used for bias analysis.*

In this way, all comments are essentially split into four groups: true positive with identity terms (TPwIT), false positive with identity terms (FPwIT), true negative without identity terms (TNwoIT) and false negative without identity terms (FNwoIT). We plot the subjectivity score distribution over false positives and true positives with box-plot diagrams. As shown on the left of Figure 4.2, for comments with identity terms, the true positives (i.e., toxic comments) show higher subjectivity levels than the false positives (i.e., non-toxic comments) across all tasks. First, false positives have a lower median of subjectivity scores than true positives across four datasets. Second, the false positive predictions have a generally smaller and lower interquartile range than the true positive predictions in the task. The lower subjectivity scores in false positive predictions may reflect the real-word scenario that when speakers talk about a demographic group such as female, Muslim or Asian in an objective way, e.g., describing the group neutrally, the speech is less likely to be disrespectful or offensive. On the other hand, toxic comments often involve subjective expressions.

Notably, the pattern of lower subjectivity level of false positives is consistent only among comments with identity terms. The comments without identity terms, as shown on the right diagram in Figure 4.2 do not indicate a consistent pattern

between false positives and true positives. This shows that the feature of subjectivity level could be indicative only when considering the presence of identity terms at the same time. On the other hand, it also indicates that the identity term bias can be addressed by considering the subjectivity level of a comment. Nonetheless, this is not to assert that a text mentioning identity terms in a subjective tone should be toxic. As shown in the left boxplot in Figure 4.2, there are indeed non-toxic comments with identity terms (i.e., FPwIT in green) that have been assigned with subjectivity scores over 0.5 on the dataset 42k and Wiki. We looked into these comments and present two examples:

"mike ##pen ##ce not being able to have dinner alone with any woman other than his wife etc is just like being a strict muslim. ironic" (score of 0.63).

"ga##bs whites got nothing for free you won t it s the way life work stop w##hini##ng ab" (score of 0.8).

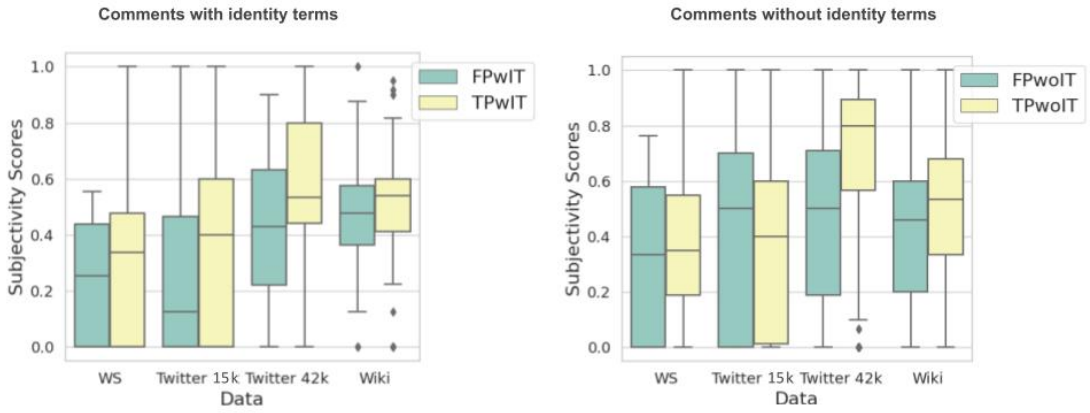

Figure 4.2: *The comparison of subjectivity level scores between true positive (i.e. toxic comments, as coloured in yellow) and false positive predictions (i.e. non-toxic comments, as coloured in green) by BERT over four TCC tasks. This figure is better viewed in colour.*

4.4.5 Subjectivity level and sentiment polarity

We also conduct an analysis of the sentiment polarity and its relation to the identity term bias. Same as our previous analysis, we use TextBlob to calculate a sentiment score for each true positive and false positive from the BERT predictions on the four TCC tasks. The sentiment score is a floating number within the range

from -1.0 to 1.0. A comment with a score close to -1 indicates a negative sentiment polarity, a score close to 1 indicates a positive sentiment polarity (Loria, 2018). As shown in Figure 4.3, there is no consistent pattern observed over four tasks regardless of their prediction results or if they contain identity terms or not. That is, the sentiment scores over the four tasks demonstrate a random distribution between true positive and false positive; and also a random distribution between comments with and without identity terms. Such results suggest that the identity term bias is hardly associated with sentiment polarity.

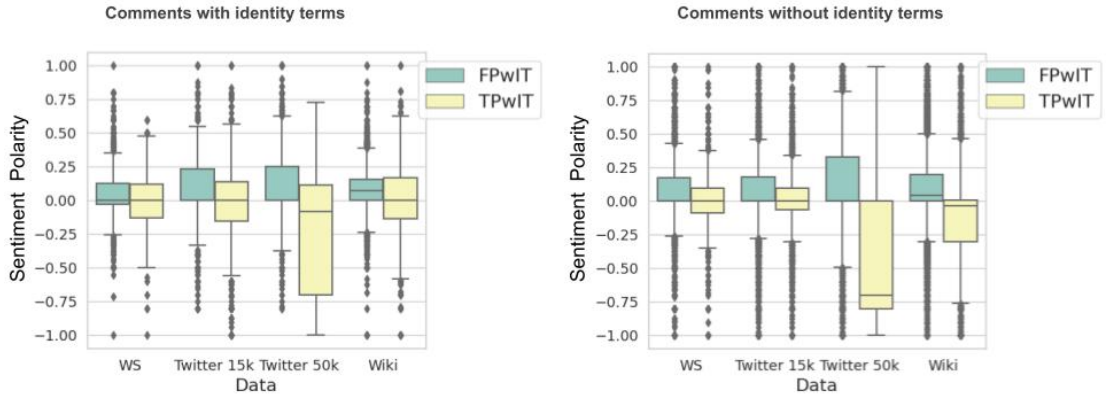

Figure 4.3: *The comparison of sentiment polarity scores between true positive (i.e. toxic comments, as coloured in yellow) and false positive predictions (i.e. non-toxic comments, as coloured in green) by BERT over four TCC tasks. This figure is better viewed in colour.*

4.4.6 Summary

In summary, our study has shown that BERT over-attends to identity terms when making predictions. This over-attention on identity terms can contribute to false positive predictions. Furthermore, using the predictions by a BERT classifier, we observe that when identity terms are present, false positives tend to have lower subjectivity scores while true positives generally have higher subjectivity scores.

4.4.7 Limitations

An issue that was not addressed in this study was the absence of a clear definition of subjectivity and different levels of subjectivity. As mentioned above, to the best of our knowledge, no previous study has provided a clear definition of

subjectivity in the context of toxic comment classification or related tasks. This might lead to a debate of our findings. A clear definition of subjectivity is also important to communicate with the public when applying our findings to a real-life scenario. Therefore, we believe it will be a meaningful direction for future studies that define subjectivity and different subjectivity levels in the context of toxic comments. The second limitation is that we only tested BERT. Although it is the most representative PLM, other advance PLMs proposed later might demonstrate different patterns. Another limitation lies in the fact that TextBlob uses a simple lexicon-based approach, which can produce inaccurate subjectivity scores for some comments. Last, a concern raised by this study is the list of identity terms used in the experiment. Previous TCC studies of the identity term bias have not drawn on any systematic research into defining identity terms. Research up to now has been empirical on selecting identity terms. Although certain terms such as "Muslim" and "black" are easy to decide, terms such as "heterosexual" and "girls" can be controversial in different scenarios. Several related problems are still open: should different TCC tasks use different identity terms? How to evaluate the list has covered a proper range of different groups? How to handle identity terms that has many variants names (e.g., slang and acronym)?

4.5 Conclusion

This chapter conducts a deep investigation into the identity term bias in one SOTA TCC classifier, BERT. We investigated the model explanations and conducted both qualitative analysis and quantitative analysis on four different TCC tasks. From the model rationales, we observe the evidence across all tasks, showing that BERT over-attends to identity terms during the model decision-making process. This has addressed the first and the second research questions for this chapter: we have shown that the identity term bias is prevalent in different TCC tasks, and model rationales can provide evidence of this issue. From the predictions, we observed that when identity terms are present, false positives tend to have lower subjectivity scores while true positives generally have higher subjectivity scores. However, this pattern did not exist among the false positives without identity terms. This has addressed the third research question on the relationship between the subjectivity level and the identity term bias. We also believe this relationship can be utilized to mitigate the identity term bias in TCC tasks. We introduce our method to leverage this in the next chapter.

	Tasks	Comments	Rationales by gradients	Rationales by attentions
1	WS	there is no doubt the chinese and japs have giving the world culture.	japs chinese	japs chinese the
2	WS	like was posted above blacks are waking up more whites then prowhites.	prowhites blacks waking	prowhites waking up
3	WS	i have always thought whites and negroes are as different as lions and tigers.	negroes whites	negroes whites
4	WS	we are not filled with all the hate that the jewish media tells the lemmings we are.	jewish lemmings . hate	lemmings jewish the we
5	WS	yes and if i were a jewish - supremacist i'd want it that way, too.	jewish supremacist i	supremacist jewish a
6	WS	the reason they are killing themselves is because our weak liberal goverment are too afraid to make them do any sort of training..	goverment liberal weak	goverment they liberal them themselves
7	WS	yeah you can count on it, there will be lot of people on the streets that day protesting against this jew government what we have here.	jew protesting government yeah this	jew government this we streets yeah
8	WS	i agree the defence is legitimate, 100%. but the taking up of arms against and killing of other whites is the very last thing we need to succeed in our cause.	whites killing the the	whites of other arms we
9	WS	only 6 corporations control over 90 % of the television channels these corporations are anti - white and are turning our people into zombies let's break the monopoly of news and information by supporting alternative, pro - white media like stormfront.	zombies corporations corporations white white stormfront the the	zombies white white stormfront corporations into are people corporations
10	Twitter 18k	@ nndabbour64 mohammed's first wife khatya was his boss and was a property owner before she married mohammed and before islam.	@ . khatya ' and	nndabbour64 mohammed wife @ mohammed
11	Twitter 18k	if u call me a feminazi i will go feminazi on ur ass because ppl like u are the reason we need feminism	the ur go a i	feminazi feminazi feminism are ppl
12	Twitter 18k	@ russell _ barth @ angry _ feminazi @ liamfinn19 @ pattyarquette that does not include overtime.	feminazi @ liamfinn19 @	feminazi liamfinn19 pattyarquette overtime
13	Twitter 18k	i was just called a " feminazi " which is sadly not unusual on the show	a feminazi	feminazi show i the
14	Twitter 18k	nigerian military kills 104 boko haram islam terrorists. egyptian military kills 47 isis siani islam terrorist	siani islam	nigerian islam kills haram
15	Twitter 18k	a girl thats rly nice and cute just shared a post from anti - feminazi on fb. i had so much faith in u. u	thats i and	feminazi girl i fb rly thats
16	Twitter 18k	if u use the term feminazi u can just go ahead and stay away from me	the can feminazi just	feminazi the me away
17	Twitter 42k	rt la2ca women who claim they don t get wifed because they aren t hoes are just as bad as dudes who claim they re overlooked bc the	the wifed dudes hoes rt la2ca	women rt hoes wifed dudes la2ca
18	Twitter 42k	college doesn t sound very good with all your lousy liberal brainwashing thanks 2 u they hate us law history amp truth	lousy amp hate brainwashing thanks	hate law they lousy liberal
19	Twitter 42k	we will see what happens. i just hope this does not cause muslims hate. extremist if they did it	i extremist muslims	muslims extremist hate
20	Twitter 42k	celinedoux 22yo camgirl petite body 35b tits brown eyes and brown hair. womancam gt gt	tits and celinedoux	tits celinedoux womancam
21	Twitter 42k	lt lesbian chronicle late at the office part 1 she ran her hands over mariam s breasts	breasts lesbian the s	lesbian breasts mariam s
22	Wiki	mcc or metropolitan community church is a church for all people who believe in our lord jesus christ. a gay lesbian's bisexuals, transgengers, straight people. a church that never descreimanate, whoever, whatsoever you are, we are all created by god equal.	you bisexuals lesbian transgengers for gay	bisexuals gay lesbian is or god a
23	Wiki	i agree. it is entirely logical that a person who enjoys the company of homosexuals is a homophile. a group of supporters and friends of homosexuals - who are not themselves homosexuals - can (should) be called homophile. but an anglophile is someone who likes england, but isn't english. a group made up predominantly of homosexuals cannot be called homophile, applying the same logic. it can only be called homosexual.	homosexuals homosexuals homosexuals homosexuals homosexual homophile homophile homophile i the the	i homosexuals homosexuals homosexuals anglophile homosexuals agree entirely homosexual
24	Wiki	hi i'm angel of sadness i'm a 10 years old girl that love gay band tokio hotel and you?	gay i i m m	gay love
25	Wiki	ok, i got it that not all gay mean speak this way but is it really necessary to have 50 % of the article stating that this is a stereotype and that some (but not all!) homosexuals speak this way?	gay the homosexuals it it ok	homosexuals gay i mean but ok
26	Wiki	fyi i am a gay man, if anyone wants to chat just comment plz	fyi gay	fyi plz gay

Table 4.2: *False-positive prediction examples for those whose rationales contain identity terms. Punctuations are removed from rationales.*

Identity Terms	Predictions	Comments	Subjectivity scores
muslim	True Positive (toxic)	i have been talking about it for years. the filthy **muslims** putting their sight on africa. in some ways they are worst then blacks.	1
	False Positive (non-toxic)	@ kelly ##turn ##er ##9 ##9 @ bu ##eller ##ish ##ere more **muslims** die in terrorist attacks than anyone else. for example, 80% of afghan civilians killed by taliban	0
gay	True Positive (toxic)	if this guy isn ##t gay he is one seriously good actor !!!!! i've seen hayes on will & grace and the way he acts so.... feminine he must be homosexual. no straight man can act so convincing ##ly **gay**. but then again sean acted pretty straight and normal in the movie where he played jerry lewis. my opinion is still in favour of him being gay though.	0.6444
	False Positive (non-toxic)	f ##yi i am a **gay** man , if anyone wants to chat just comment pl ##z	0.3417
whites	True Positive (toxic)	black people in sa do not want to unite with **whites** to remove corruption . we ' d rather endure rot & amp; suffer! so sad!	0.6444
	False Positive (non-toxic)	i' m reporting here that i haven' t seen or heard anything on talk radio or tv that even hints at jewish su ##pre ##mac ##ist ethnic cleansing of **whites** from usa or europe .	0
women	True Positive (toxic)	rt @ mw##il##bu##r##6##2 i swear i am not sex ##ist , but **women** ' s tennis is the most boring sport to watch	0.75
	False Positive (non-toxic)	men and **women** are segregated in mosques .	0
jews	True Positive (toxic)	well i am sure that the **jews** have the cure for aids but they won ##t give it to the fa ##gs cause they already served their pu ##rp ##oise	0.8888
	False Positive (non-toxic)	if a third world savage attacks a **jew**, is it all jew ##ey ' s fault?	0

Table 4.3: *Example comments that contain identity terms. Each pair of comments contain the same identity terms, while one is true positive and the other is false positive predicted by BERT.*

Chapter 5

Mitigate the identity term bias

As studied in the last chapter, toxic comment classification models are often found to suffer from identity term bias. Instead of training a model to pay less attention to identity terms, we propose a different approach in this chapter. Built on the findings from the last chapter, our approach leverages the notion of subjectivity level of a comment and the presence of identity terms. We hypothesise that when a comment contains an identity term that refers to a particular group of people, the likelihood of that comment being toxic is associated with the subjectivity level of the comment, i.e., the extent to which the comment conveys personal feelings and opinions. Therefore, this chapter introduces a novel method that incorporates this idea to mitigate the identity term bias on TCC tasks.

The remainder of this chapter is structured as follows. Section 5.1 first gives a brief overview of our research background and research questions. Section 5.2 reviews previous methods of handling the identity term bias. Section 5.3 presents the methodology used for this chapter. Section 5.4 explains our experiment details and discuss the results. Section 5.5 concludes this chapter and proposes potential research directions.

5.1 Introduction

Previous work have identified the identity terms bias in TCC models. Limited studies have attempted to handle such bias and those methods can be categorised into two groups: debiasing the dataset and debiasing the model (Dixon et al., 2018; Davidson et al., 2019; Wiegand et al., 2019). The former group aims to mitigate the identity term bias via modifying the dataset. The intuition is removing bias related features from the dataset directly. The later group, debiasing the model, aims to

modify the model itself to mitigate the bias. There are two main approaches of debiasing the model by previous studies: using ensemble models and adding regularisation terms (Vaidya et al., 2020; Halevy et al., 2021; Clark et al., 2019; Zhang et al., 2018a; Prost et al., 2019; Xia et al., 2020; Mozafari et al., 2020; Kennedy et al., 2020). These two approaches follow a simple principle: ignoring or paying less attention to the identity terms. However, this overlooks the fact that identity terms can be essential and important features to make predictions.

This chapter explores a new approach, which is based on the hypothesis discussed earlier. A comment with a low subjectivity level expresses more factual information and less personal feelings and opinions; while a comment with a high subjectivity level contains more personal opinions but less factual information. Intuitively, when someone especially discusses a certain group of people, we pay more attention to the overall subjectivity level of the whole comments regarding the group rather than ignoring or paying less attention to that group the speaker refers to. We hypothesise that when a toxic comment is made about a group of people with an identity term, it is more likely to have a high level of subjectivity. Therefore, the likelihood of a comment being toxic can be associated with 1) whether the comment contains an identity term, and 2) the subjectivity level of that comment.

Building on the BERT model which is commonly used for multiple downstream NLP tasks, we propose a novel structure, Subdentity-Sensitive BERT (SS-BERT), where "Subdentity" denotes "subjectivity" and "identity". SS-BERT makes use of the special embedding structure of BERT to "activate" the subjectivity features only when the comment contains identity terms. In measuring the subjectivity level, we study two options. One is based on a SOTA tool for calculating subjectivity of a text using pre-defined lexicon. The other is based on the idea of calculating the similarity between a comment containing an identity term and the Wikipedia summary text (to be defined later) regarding the identity term, and using the similarity degree as a proxy to the "subjectivity". Last but not least, we also adapt our method to RoBerta to further validate the generalisation of our method.

We answer the following research questions:

- First, how to create SS-BERT that effectively incorporates the subjectivity level in BERT given the presence of the identity terms?
- Second, how to measure the subjectivity level of a comment and make it usable by SS-BERT?

- Third, to what extent can SS-BERT benefit from the subjectivity level and the presence of the identity term with respect to mitigating the identity term bias found on TCC tasks?

We compare our proposed methods against SOTA baselines (BERT and BERT+SOC (Devlin et al., 2019; Kennedy et al., 2020)) and a few alternative models designed for ablation analysis. We evaluate all models and methods on a wide range of TCC tasks. The results shows that our method is able to mitigate the identity term bias and improve toxic comment classification effectively. First, SS-BERT consistently outperforms BERT, regardless of how subjectivity is measured. The performance gain of SS-BERT is mainly attributed to predicting fewer false positives. This indicates that our idea of considering subjectivity and the presence of identity terms is helpful to mitigate the false positive bias, i.e., the identity term bias. Second, SS-BERT consistently outperforms its alternative model SO-BERT(Subjectivity-Only BERT), which only uses subjectivity without considering the presence of identity terms. This indicates that simply learning the subjectivity level for all comments is not enough, and it is more informative to combine subjectivity with the presence of identity terms. Third, our Wikipedia-similarity based proxy to subjectivity is shown to be more effective than the SOTA lexicon-based tool for TCC as models based on this measure of subjectivity have outperformed their counterparts on 3 out of 4 tasks. This suggests that given a comment containing an identity term, comparing the meaning of that comment against a reference text describing that identity term can, to some extent, reflect the subjectivity level of the comment.

5.2 Related work

Recent studies show that TCC classifiers often suffer from unintended bias and the identity term bias is one kind of such unintended bias (Park et al., 2018; Sap et al., 2019; Davidson et al., 2019; Zhou et al., 2021; Field et al., 2021; Halevy et al., 2021). TCC research has attempted to address unintended bias in two different perspectives: biases associated with datasets and biases associated with the model (Mozafari et al., 2020; Wiegand et al., 2019). Biases associated with datasets mainly refer to the biases introduced by the data labelling or the collection process (Dixon et al., 2018; Davidson et al., 2019; Wiegand et al., 2019). Biases associated with the model refer to the biases learned by the model in the training process. **Debiasing the dataset** and **debiasing the model** are two different

but non-exclusive approaches to handle the unintended bias on TCC tasks. The former one essentially curates the data to remove the bias from the dataset, while the latter designs algorithms to mitigate the bias from the model.

Wiegand et al. (2019) investigate six TCC datasets to study why TCC datasets contain biases and the bias degree of different TCC datasets. For instance, they find that over 70% of tweets labelled as sexist originate from two Twitter authors, which can contribute to the bias. Research on **debiasing the dataset** modifies the dataset, implicitly assuming there are bias-related features in the dataset that can be removed or reduced. Dixon et al. (2018) debias TCC datasets by adding non-toxic comments with identity terms to "balance" the training data, intuitively allowing the model to learn more features of non-toxic comments with identity terms. Vidgen et al. (2020) create "balanced" TCC datasets following an iterative procedure. The main idea is to let human annotators manually create data that are challenging for the model to predict correctly. This kind of data, also known as adversarial data or perturbations, manipulates the original text just enough to flip the label (e.g., from "Hate" to "Not Hate") (Kaushik et al., 2019; Gardner et al., 2020). Adversarial data is believed to enrich current datasets that are created in potentially biased ways. For example, some hate speech datasets are created by collecting tweets containing hate-related keywords or containing certain hashtags, and some datasets are annotated by a limited group of annotators (Waseem, 2016b; Wiegand et al., 2019; Fortuna and Nunes, 2018). In their method, the model is trained on a new dataset that contains half adversarial data and half "originally entered content" (i.e., the collected real data). The prediction results of the current round are then used to guide human annotators to provide a new round of adversarial data. This process is repeated for four rounds. Therefore, more new adversarial data is provided by human annotators in each round to train a new model. Vidgen et al. (2020) have found that the model trained on the data from all four rounds performs better than all the other models that are trained on non-adversarial data or trained on different rounds of adversarial. However, they also acknowledge the potential annotator bias in their method.

Interestingly, there is an emerging debate between "curating data" and "studying the world as it is" in the NLP research community (Bender and Friedman, 2018; Bender et al., 2021; Buckman; Rogers, 2021). Those favouring "curating data" argue that the real data reflects the real world which has discrimination and curating data is a method to combat discrimination against different social groups (Blodgett et al., 2020; Bender et al., 2021). Another supporting argument is that machine learning models may memorise specific facts which can expose person-

ally identifiable information. Curating data is a way to remove these identifiable personal information for ethical reasons. What is more, deep learning models are vulnerable to basic perturbations and attacks, such as adversarial data designed by humans targeting on the specific bias of the model. This can be "solved" with shallow data curation (Ribeiro et al., 2020). However, those favouring "studying the world as it is" argue for algorithmic solutions to address similar issues mentioned above (Clark et al., 2019; Rogers, 2021; Ribeiro et al., 2020; Garrido-Muñoz et al., 2021). Also, they argue that curation means making conscious choices about what to include and what to exclude and this raises new questions: what is the standard for it, and what is the proper degree of curation (Rogers, 2021). Particularly, Sambasivan et al. (2021) suggest that "conventional algorithmic fairness is west-centric". Changing the data possibly inserts new values and the "west-centric" values which interpret the histories and cultures of non-western societies from a Western perspective can be controversial[1] (Amin, 1989; Blaut, 1993; Wallerstein, 1997). Last, Lissack (2021) argues that supports of curating data are "advocacy rather than research" (Rogers, 2021). We would like to direct interested readers to papers by Rogers et al. (2021), Bender and Koller (2020), Garrido-Muñoz et al. (2021), Rogers (2021) and Sambasivan et al. (2021) for further details.

In contrast to debiasing the dataset, there are more studies on **debiasing the model**. Two popular approaches are introduced in the literature: using ensemble models and adding regularisation terms. In short, the idea of using ensemble models involves adding an additional classifier to learn the bias-related features and letting the main classifier learn bias-free features. In contrary, the idea of adding regularisation terms is adding additional training objectives to penalise the bias-related features (Vaidya et al., 2020; Halevy et al., 2021; Kennedy et al., 2020; Clark et al., 2019). For an example of ensemble models, Clark et al. (2019) first train a "bias-only" model on a dataset whose data has been added with deliberate information, such as indicator features. These deliberate information can be taken as artificially inserting bias into the dataset. They then train a second model in an ensemble with the pre-trained "bias-only" model on the original dataset[2]. The

[1]Eurocentrism, also known as west-centric, refers to a discursive tendency to interpret the histories and cultures of non-European societies from a European (or Western) perspective (Amin, 1989; Blaut, 1993; Wallerstein, 1997). Common features of Eurocentric thought include: ignoring or undervaluing non-European societies as inferior to Western; ignoring or undervaluing what Asians or Africans do within their own society or seeing the histories of non-European societies simply in European terms, or as part of "the expansion of Europe" and its civilizing influence (Amin, 1989; Blaut, 1993; Wallerstein, 1997).

[2]Their method involves changing the dataset. However, we group it into debasing the model as their final model is only trained on the original dataset.

intuition behind is to allow the second model to learn less bias-related features as the "bias-only" model "absorbs" the bias-related features.

With respect to adding regularisation terms, more previous work includes (Zhang et al., 2018a; Prost et al., 2019; Xia et al., 2020; Mozafari et al., 2020; Kennedy et al., 2020). Kennedy et al. (2020) find that BERT neglects the context around the identity terms, which has led to many false positives. They thus propose BERT+SOC (Sampling and Occlusion), which is built atop BERT with an extra regularisation term to predict an "importance score" of identity terms. The idea is to minimise the prediction differences between when an identity term is exposed to the model and when it is hidden from the model via the importance scores. Ideally, the "over-attended" identity terms will be assigned with low importance scores and thus they will become less indicative of whether the comment is hate speech or not. Therefore, their method encourages the model to pay less attention to identity terms, which can sometimes be actually useful features. Similarly, Prost et al. (2019) add a regularisation to penalise the dependence between the distribution of predicted probabilities and protected features, such as the dependence between toxic labels and identity terms. That is, the model attempts to minimise the prediction difference between protected features and other non-protected features. Another example is provided by Mozafari et al. (2020) who use an regularisation term to re-weigh input samples to suppress the effect of highly correlative n-grams found in the training set. Studies that follow a similar direction include Zhang et al. (2018a) and Xia et al. (2020).

Zhou et al. (2021) compare the performance of debiasing the dataset and debiasing the model on one TCC task. In order to debias the dataset, they filter out "too easy" data that might contain spurious correlations or biases. They test different methods to find those "too easy" instances. For example, one method is AFLite proposed by Le Bras et al. (2020) and its intuition is that examples predicted correctly by the simplest methods likely exhibit spurious biases. For debiasing the model, they insert additional training objectives to the model by adapting the method by Clark et al. (2019). They have found that models trained on debiased datasets (i.e., debiasing the dataset) achieve overall higher performance than models with additional training objectives (i.e., debiasing the model), while the latter performs better on lexical bias reduction. Here, lexical bias refers to the bias towards a list of "bad words", also known as "Toxicity Triggers" in their paper. However, Zhou et al. (2021) have not compared their approach against other SOTA methods that debias the model, such as adding an regularisation proposed by Kennedy et al. (2020) and a multi-task-based method proposed by Vaidya et al.

(2020). Another limitation is that Zhou et al. (2021) have experimented with only one task and this limits the generalisability of their findings.

In summary, the identity term bias has become a focus of TCC study in recent years. Existing approaches to addressing this bias either debias the data, or the model. Methods of debiasing the dataset usually modify the dataset to make it more balanced. However, it is unclear how such an approach can be generalised and transferred to a different task or domain. Methods of debiasing the model mostly follow a similar principle that encourages the model to ignore or pay less attention to identity terms. However, this overlooks the important fact that in particular situations, such terms are useful for prediction. As an example, given the sentence "women cannot drive" and the sentence "children cannot drive", the identity terms "women" and "children" are crucial in correctly classifying the sentences. Ignoring these terms may lead to false prediction. In this example, "children cannot drive" expresses a common sense rather than disrespect or hate towards children, while "women cannot drive" can be sexist.

In this work, we explore a new venue for debiasing TCC models, which take use of the presence of identity terms in the comment and the subjectivity level of the comment. The novelty is that we consider identity terms together with the extent to which a message expresses subjective opinions (we refer to this as "subjectivity level" in the following). We demonstrate this with an in-depth analysis in the next section. Additionally, we propose that the semantic similarity between a comment and an "objective" reference text can be used as a proxy to measure subjectivity in our model. Specifically, given a comment containing an identity term, we compute the similarity between the comment and a Wikipedia description of the identity term and use the similarity as a proxy to the subjectivity level of the comment.

5.3 Methodology

We propose a BERT-based model that make use of the structure of BERT embeddings to add the information of subjectivity and the presence of identity terms. The design drive is to enable the model to pay attention to the subjectivity of a comment when the comment contains identity terms. When the identity terms are not present, the model should not consider the subjectivity. To do so, in short, we append an additional "token" to the end of the token sequence for each comment to "notify" the model of the information of the subjectivity level and identity terms existence. The following sections will explain this in detail regarding the model construction and the subjectivity scores respectively.

5.3.1 The task and datasets

Following the methodology outlined in Chapter 4, we study the problem in the setting of binary classification tasks and use the same TCC datasets as shown in Table 4.1 in Section 4.3.4. Each dataset is split to training, validation and test datasets (80%, 10%, 10%). The results are reported on the testing dataset.

5.3.2 SS-BERT model structure

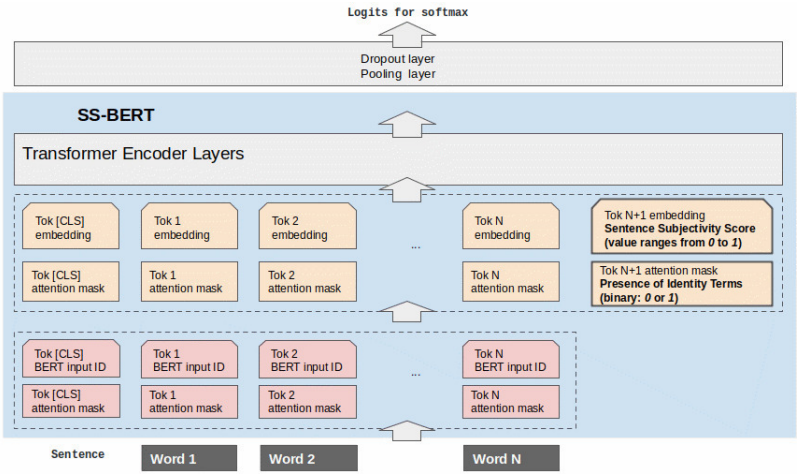

Figure 5.1: *Illustration of a SS-BERT model on classification tasks. This figure is better viewed in colour.*

As shown in Figure 5.1, building on BERT, we append an additional "token" to the end of the token sequence for each comment. We use the subjectivity score for the embedding of the token. To be more specific, we create a 3-D tensor with the same size of other token embeddings and each element's value in the tensor is equal to the subjectivity score. In the scenario of BERT, the dimension size is 768 and thus the tensor for the added "token" is a 3D-tensor of size [batch size, 1, 768]. The tensor is denoted as "Sentence Subjectivity Score" in Figure 5.1.

For the corresponding attention mask (highlighted in yellow with bold borders in Figure 5.1), we set it to indicate the presence of identity terms so that if there is no identity term in the comment, the appended "token" will be masked. While if there is an identity term, the embedding of this "token", i.e., the subjectivity score, will be attended by BERT.

5.3.3 Subjectivity score

We explore two different methods of measuring the subjectivity level of a comment. The first method is simply using the TextBlob library as mentioned before. The second is based on measuring the semantic similarity between the comment and a related Wikipedia description for the corresponding identity term found in the comment.

TextBlob

We use TextBlob as described in Section 4.3.4 to assign a subjectivity score to each comment. The subjectivity scores range from 0.0 to 1.0 where 0.0 is very objective and 1.0 is very subjective. TextBlob uses a lexicon-based approach to compute subjectivity scores. It emphasises the impact of individual words (Sebastiani and Esuli, 2006; Loria, 2018). To be more specific, TextBlob uses a vocabulary and each word in the vocabulary is associated with a subjectivity score. In the case of polysemous words, it returns the average subjectivity scores over all the possible senses of that word. For out-of-vocabulary words, TextBlob assigns a subjectivity score of 0 to the word. A comment's final subjectivity score is the mean subjectivity score of all its words.

Wikipedia based similarity

Inspired by Zhang and Yu (2006); Zhang et al. (2007); Kittur and Kraut (2008); Mesgari et al. (2015), we make an assumption that the summary section of a Wikipedia article regarding an identity term is a relatively objective description of the identity term, and the similarity between the Wikipedia summary of a given identity term and a comment mentioning the same identity term can reflect the subjectivity level of the comment. The more similar they are, the more objective the given comment is. Zhang and Yu (2006); Zhang et al. (2007) have made similar assumptions that "all the contents of these pages (Wikipedia pages) are assumed to be objective". In their work of opinion retrieval (i.e., finding relevant blog documents containing opinionated content for a given query topic), Zhang et al. (2007) compare the vectors' similarity between given documents and related Wikipedia articles to find the opinionated content.

We propose to take the cosine similarity between a given comment and a Wikipedia summary text regarding an identity term as a proxy to the subjectivity of the comment. Specifically, given a comment with a certain identity term, e.g., "muslim", "islam" and "lesbian", we retrieve the first section (i.e., the "summary

text") of the corresponding Wikipedia article of the identity term. Then, the summary is fed to BERT which gives an embedding for this summary[3]. Second, we apply the same process to the given comment to create its corresponding BERT embedding. We pool[4] both embeddings (one for the Wikipedia summary and the other one is for the given comment) to one dimension of size 768 (the hidden size of BERT) and calculate their cosine similarity ($\frac{u \cdot v}{\|u\|\|v\|}$).

The higher value of a cosine similarity, the more similar the comment is to the Wikipedia summary. Since the Wikipedia summary is generally an objective narrative of the topic (further discussed below), we hypothesize that this degree of "similarity" may capture the degree of "objectivity" to some extent. Therefore, the subjectivity of the comment, which is the opposite to objectivity, is calculated as:

$$1 - \frac{u \cdot v}{\|u\|\|v\|} \tag{5.1}$$

where u and v denote the two embedding vectors. For example, using this Wikipedia based subjectivity measure, the subjectivity score for "women cannot drive" is 0.4011 and for "children cannot drive" is 0.2937. These scores indicate the former has a higher subjectivity score, which is consistent with the discussion in Section 5.2 regarding these two sentences. This suggests that the Wikipedia summaries may have provided useful background information that helps the interpretation of the two messages that are only one-word different.

We acknowledge that this score is not a direct reflection of the subjectivity level, however, we argue that this to some extent, reflects the subjectivity level of a comment. Although Wikipedia articles are also user-generated content, their collaborative authoring nature reduces the risk of creating subjective content. Additionally, Wikipedia articles are scrutinized by a wide community which helps ensure the descriptions regarding identity terms are accurate and objective (Kittur and Kraut, 2008; Mesgari et al., 2015). In particular, articles about identity terms are edited many times by different contributors and reviewers from different backgrounds (Hu et al., 2007). Also, earlier studies showed that the summary section of a Wikipedia article usually defines the topic in question (Ye et al., 2009; Sankarasubramaniam et al., 2014). To sum up, the summary text of an Wikipedia article of a certain identity term can be reasonably expected to be objective de-

[3]We constrain the maximum length of the summary to be 500 tokens. The 25 embeddings (one embedding for one summary of its identity term) are saved as a lookup table to speed up the training process.

[4]Here, each embedding is a 2D tensor with a size of [token length, 768]. 768 is the hidden size of BERT. We average the first dimension which gives a 1D tensor with a size of [768].

scriptions of the topic. We show empirically later in the experiments that this is useful for our model. We call this method "Wikipedia based subjectivity" and denote it as *(Wikipedia)* when a model uses this score as subjectivity scores, i.e., SS-BERT (Wikipedia) and SS-BERT+SOC (Wikipedia).

Since SS-BERT does not attend to the subjectivity level of comments without identity terms and we cannot compute the Wikipedia based subjectivity for such comments, their subjectivity will be assigned as zero. For comments with multiple identity terms, we repeat the above process for each identity term and use the mean value as the final subjectivity score.

5.4 Experiments

5.4.1 Comparative models, datasets and implementation

We design two baseline models and two variations of our SS-BERT, in order to fully evaluate the effects of our design of SS-BERT. For SS-BERT and the two variations, we experiment with the two different ways of calculating subjectivity as detailed in Section 5.3.3.

Baselines The first baseline is an initial BERT, as explained in Section 4.3.2. The second baseline, BERT+SOC, is an implementation of the SOTA method in Kennedy et al. (2020), as described in Section 5.2. To the best of our knowledge, Kennedy et al. (2020) is the only work that focuses on mitigating identity terms bias found in BERT. Additionally, we show the result of using heuristics (rule based) as a reference. It simply predicts any instance containing one or more identity terms with a subjective level higher than 0.3 (RB-3), 0.5 (RB-5), or 0.8 (RB-8), as positive ("toxic").

Subjectivity-Only BERT (SO-BERT) To examine if the subjectivity level is generally helpful for TCC tasks regardless of the presence of identity terms, we create a variation of our method, SO-BERT, which only captures the information of subjectivity level but not the presence of identity terms. To do so, we adapt from the SS-BERT model with the attention mask always attending the added "token". The rest of the model structure remains the same. We only test subjectivity scores by TextBlob for this comparative model, denoted as "SO-BERT (TextBlob)", as our Wikipedia based subjectivity method cannot calculate the similarity without the presence of identity terms (which are used to look up the corresponding Wikipedia articles for comparison). Note that the SO-BERT is different from the previous TCC works incorporating subjectivity in their methods as

Data	RB-0	RB-3	RB-5	RB-8
WS	0.3253	0.2379	0.1656	0.1656
Twitter 18k	0.3985	0.2286	0.1510	0.1510
Twitter 42k	0.0807	0.0707	0.0521	0.0521
Wiki	0.0950	0.0884	0.0596	0.0596

Table 5.1: *The prediction F1 results of using rule-based heuristics. RB-3 represents a rule based threshold of the subjective level threshold for labelling "toxic" is 0.3. Similar rules apply to RB-0, -5 and -8.*

SO-BERT embeds the subjectivity level in a pre-trained LM directly rather than using it prior to the model, such as the work by Gitari et al. (2015) as described in Section 4.2, or as feature in a traditional machine learning fashion.

Subjectivity-BERT+SOC (SS-BERT+SOC) combines the method of BERT+SOC by Kennedy et al. (2020) and our model (SS-BERT) to create a hybrid BERT-based model, SS-BERT+SOC. In short, it learns the subjectivity level and the presence of identity terms with the added "token" and also has an extra regularisation term in its loss function which encourages the model to learn more from the context of the identity term and less from the identity term. We experiment with the two different subjectivity measures detailed in Section 5.3.3, i.e., SS-BERT+SOC (TextBlob) and SS-BERT+SOC (Wikipedia).

5.4.2 Results: F1 comparison

Data	Data Size	Baseline Models				SS-BERT			
		BERT		BERT+SOC		SS-BERT(TextBlob)		SS-BERT(Wikipedia)	
		F1	std	F1	std	F1	std	F1	std
WS	10,703	0.5811	0.0204	0.5885	0.0209	0.5952	0.0203	**0.5970**	0.0175
Twitter 18k	18,625	0.7780	0.0204	0.7780	0.0055	**0.7804**	0.0080	0.7803	0.0052
Twitter 42k	42,314	0.7637	0.0071	0.7643	0.0101	0.7683	0.0059	**0.8000**	0.0081
Wiki	159,571	0.7680	0.0175	0.7548	0.0135	0.7693	0.0086	**0.7735**	0.0086

Table 5.2: *The comparison of F1 between SS-BERT and baseline models on the 4 TCC tasks. The mean F1 score and its standard deviation are from 10 independently runs for each model presented.*

SS-BERT We first analyse the F1 performance of SS-BERT. Table 5.2 shows F1 scores of the two baseline models and the two SS-BERT models with different subjectivity measures on four datasets. Overall, both SS-BERT models outperform the two baseline models consistently across four datasets. Table 5.1 shows F1 scores of using a simple heuristic (rule-based) model, indicating the weak performance of using a heuristic model.

Between the two baselines[5], BERT+SOC is able to improve BERT on WS which reflects the results reported in Kennedy et al. (2020). However, the F1 score is only marginally higher on Twitter 42k and remains the same on Twitter 18k, but decreases on Wiki compared to BERT. The under-performance of BERT+SOC on Wiki may indicate that BERT could have benefited from training on the significantly larger dataset (compared to WS, Twitter 18k and Twitter 42k) such that the extra learning objective enhancing the contextual information around identity terms may have had negligible influence on the model. In contrast, our two SS-BERT models outperform BERT on all datasets. This suggests the mechanism of attending to subjectivity based on the presence of identity terms cannot be compensated by dataset size. Therefore, the results show that our model brings unique benefits and that is the reason SS-BERT still outperforms BERT on Wiki.

Between the two different subjectivity measures, SS-BERT (TextBlob) works better on the two smaller datasets, WS and Twitter 18k, while SS-BERT (Wikipedia) works better on the two bigger datasets, Twitter 42k and Wiki. A possible explanation for this might be that the average comment length of Twitter 42k (123) and Wiki (398) are longer than WS (91) and Twitter (96) 18k. They are in fact, of more similar lengths to the Wikipedia summary text. A short comment intuitively may contain less semantic information and may not provide sufficient features for the similarity computation. Another possible explanation for this is that the Wikipedia summary text has provided background information regarding identity terms in addition to the target datasets.

Variation models using subjectivity We compare our method SS-BERT against the other two variation models which also utilize subjectivity scores, namely SO-BERT and SS-BERT+SOC. Table 5.3 shows models using subjectivity scores by TextBlob on the top and those by models using Wikipedia based subjectivity at the bottom. The model under-performing one of the baselines shown in Table 5.2 are enclosed in parentheses "[ ]" and models under-performing both baselines are enclosed in parentheses "[[ ]]".

Comparing SS-BERT against SS-BERT+SOC, we notice the following patterns. When using TextBlob for measuring subjectivity, SS-BERT consistently achieves the best performance on all tasks. When using Wikipedia based subjectivity, SS-BERT noticeably outperforms SS-BERT+SOC on the two bigger datasets

[5]We notice that our results of BERT and BERT+SOC on the WS dataset are different from that reported in Kennedy et al. (2020), as our F1 are higher. While our results are obtained by re-running their code as-is, a possible reason for this difference is that the only version of the data we can download has been modified from that used in the authors' original study.

Data	Data Size	SS-BERT		Comparative Models with Subjectivity			
		SS-BERT(TextBlob)		SS-BERT+SOC(TextBlob)		SO-BERT(TextBlob)	
		F1	std	F1	std	F1	std
WS	10,703	**0.5952**	0.0203	0.5912	0.0216	0.5909	0.0247
Twitter 18k	18,625	**0.7804**	0.0080	0.7785	0.0050	[[0.7774]]	0.0055
Twitter 42k	42,314	**0.7683**	0.0059	0.7660	0.0056	[0.7636]	0.0061
Wiki	159,571	**0.7693**	0.0133	[0.7568]	0.0112	[0.7654]	0.0151

Data	Data Size	SS-BERT		Comparative Models with Subjectivity	
		SS-BERT(Wikipedia)		SS-BERT+SOC(Wikipedia)	
		F1	std	F1	std
WS	10,703	0.5970	0.0175	**0.5980**	0.0272
Twitter 18k	18,625	0.7803	0.0052	**0.7812**	0.0036
Twitter 42k	42,314	**0.8000**	0.0081	0.7687	0.0068
Wiki	159,571	**0.7735**	0.0086	[[0.7539]]	0.0149

Table 5.3: *The comparison of F1 of different methods utilizing subjectivity scores on the 4 TCC tasks. The mean F1 score and its standard deviation are from 10 independently runs for each model presented. The model under-performing one of the baselines are enclosed in parentheses "[ ]" and models under-performing both baselines are enclosed in parentheses "[[ ]]".*

and achieves comparable results to SS-BERT+SOC on the two smaller datasets. It is worth noting that the F1 scores by SS-BERT-SOC on WS and Twitter 18k are only marginal higher than SS-BERT whereas SS-BERT+SOC obtains F1 that are lower than the two baselines on the Wiki dataset. Overall, SS-BERT works better than SS-BERT+SOC. A possible reason can be that the extra regulation from SOC might dilute the impact that SS-BERT brings to the model.

Comparing SS-BERT against SO-BERT, we notice adding subjectivity information to the model alone does not lead to improvement to BERT on most tasks. In other words, the model that considers subjectivity regardless of the presence of identity terms, does not consistently benefit TCC tasks. Only adding the information of the subjectivity and the identity terms presence together consistently improves over the baselines. This reflects the intuition we mentioned previously and the pattern we identified in Section 4.4.4 that the subjectivity level of a comment is an indicative feature for toxicity only if identity terms are present in the comment.

Another important finding is that although BERT+SOC is designed to mitigate the identity term bias, it is not able to learn the subjectivity level of comments with identity terms. Therefore, SS-BERT+SOC outperforms BERT+SOC consistently, suggesting that adding the subjectivity level and the presence of identity

terms can improve BERT+SOC performance.

5.4.3 Results: the identity term bias

DATA	BERT		SS-BERT(TextBlob)		SS-BERT(Wikipedia)	
	False Positive	False Negative	False Positive	False Negative	False Positive	False Negative
WS	**34**	57	**34**	55	36	54
Twitter 18k	149	117	**146**	117	149	114
Twitter 42k	227	78	214	81	**134**	101
Wiki	750	168	738	170	**608**	176

Table 5.4: *Summary of false positives and false negatives of BERT and SS-BERT on the 4 datasets. Mean values of the performance across 10 runs are shown. The lowest False Positive on each task is in bold.*

We compare the erroneous predictions between SS-BERT and the baseline BERT to further investigate SS-BERT's performance, especially regarding the model's ability to handle the false positive bias and the identity term bias. First, as shown in Table 5.4, SS-BERT is able to decrease the false positive predictions in general. The dataset Twitter 42k benefits from this the most, considering the number of reduced false positives with respect to the dataset size. This is consistent with Figure 4.2 which shows a noticeable gap of subjectivity levels between false positive and true positive predictions from BERT on Twitter 42k.

Reflecting on the improved performance on the two bigger datasets, Twitter 42k and Wiki, SS-BERT (Wikipedia) effectively reduces the false positives on these two datasets. However, SS-BERT (Wikipedia) cannot effectively reduce false positives on WS or Twitter 18k. Nevertheless, SS-BERT (Wikipedia) leads to noticeable decrease of false negatives on these two datasets. This overall leads to improvement in F1 (shown in Table 5.2) as SS-BERT (Wikipedia) maintains the same levels of false positives on the two smaller datasets.

These results may be explained by the fact that the baseline model BERT has more false positives than false negatives over the two datasets. To be more specific, 74.43% and 81.70% erroneous predictions are from false positive cases for Twitter 42k and Wiki, while the numbers are only 37.36% and 56.02% for WS and Twitter 18k. The higher false positive percentage from the baseline model indicates more room for improvements. Last, as we mentioned in Section 5.2 and identified by Wiegand et al. (2019), the dataset Twitter 18k can be problematic as over 70% of the tweets labelled as sexists were written by just two Twitter users. We also identified some arguable labels in the two smaller datasets in our

investigation.

5.4.4 Error analysis

We perform an error analysis to shed light on the limitations of the best performing model (SS-BERT) with a focus on the identity term bias. First, the identity term list we use may not "fit" the datasets well. An updated identity term list may bring further improvement. Specifically, there is only a small percentage of comments that contain the identity terms used in the model. For WS, Twitter 18k, Twitter 42k and Wiki, the ratio of comments with identity terms in the list are 21.20%, 19.90%, 4.22%, and 6.02% respectively. Therefore, only a small percentage of the datasets could have benefited from our method. For example, we observe that some false negatives and false positives contain identity terms like "liberal", "irish", "asian", and "chinese", which are not included in the identity term list used in our model. Examples are listed in Table 5.5 as shown below. Kennedy et al. (2020) uses a supervised approach combined with manual screening to select identity terms. One approach we have tried to address the issue is to create a heuristics list of identity terms that covers as many identity terms as possible which are suffering from identity terms bias. However, this attempt failed as shown in the falling performances. We believe a non-heuristics solution to create an identity term list that can improve the performance can be a future direction.

Comments	Group identity terms	SS-BERT Predictions
liberal is just the pc word for rap ##ist .	liberal	False Negative
i get what your saying and there are thousands of non irish here who will never contribute anything to this land .	irish	False Negative
i suppose the concept of a male cow and a female cow producing more cows , and the repeating process is too much for them to understand .	female	False Negative
traditionally , there have always been a lot of them despite the fact that ne asian ##s are the extreme version of our breeding strategy .	asian	False Negative
there is no doubt the chinese and ja ##ps have giving the world culture .	chinese	False Positive
the reason they are killing themselves is because our weak liberal gov ##er ##ment are too afraid to make them do any sort of training . .	liberal	False Positive
she came to hate the african po ##ache ##rs and showed no mercy towards them .	african	False Positive
these girls are not there for the cooking # mk ##r what a joke .	girls	False Positive

Table 5.5: *Examples of false positives and false negatives with identity terms that are not included in the list by Kennedy et al. (2020).*

Second, the subjectivity scores given by TextBlob are not always accurate for each comment with identity terms. We have found several limitations of its method by investigating the implementation of TextBlob. For example, they simply take the mean subjectivity scores over all words in a comment, which may be insufficient to capture the subjectivity level accurately if the comment contains a large percentage of objective words that reduce the overall subjectivity level of the comment. Also, TextBlob assigns out-of-vocabulary words a subjectivity score of 0 and this will penalise comments with many misspellings. We believe accurate subjectivity scores can potentially help SS-BERT predict them more accurately. For example, the false negative with the identity term "white":

"why white people used to say that sex was a sin used to be a mystery to me until i saw the children of browns and mixed race children popping up all around me"

has been assigned a low subjectivity score of 0.125. The presence of identity terms and the low subjective score might mislead the model to make the incorrect prediction. More examples of this kind are shown in Table 5.6.

5.4.5 Generalization

To examine the generalization of our methods, we also test our method on RoBERTa, which is another popular pre-trained LM for TCC tasks (Liu et al., 2019c; Baruah et al., 2020). RoBERTa has a similar structure with BERT where each token is given an embedding and an attention mask (Liu et al., 2019c). We add the information of subjectivity scores to the embeddings and the presences of identity terms to attention masks, as exactly how we do with SS-BERT. To provide a fair comparison, we use an initial RoBERTa as the baseline. The implementations are similar to how we implement BERT-based models that details are shown in Section 5.3. The RoBERTa model and the SS-RoBERTa model are built on our code of BERT and SS-BERT. We use the pre-trained "roberta-base" version of RoBERTa and adopt the subjectivity scores by TextBlob. The hyperparameter settings and dataset processing are the same as the experiments of BERT-based models. The results (Table 5.7) show that our method consistently improves the baseline RoBERTa across 4 datasets with the maximum improvement up to 1.29%.

It is somewhat surprising that BERT outperforms RoBERTa in our experiments as literature generally shows that RoBERTa performs better than BERT

Comments	Subjectivity Scores	SS-BERT Prediction
@ mad ##dr ##11 there is not and never has been a country called palestine, micro ##bra ##in. and that land has been the home of jews for 3000 years.	1	False Positive
buying black jelly beans for my kids... in case they re cutting up and need a chill pill. they hate them.	0.7444	False Positive
@ love egypt @ washington ##post start by reading the historian will duran ##t. his book " our oriental heritage " describes the. muslim invasion.	0.6000	False Positive
lt lesbian chronicle late at the office part 1 she ran her hands over maria ##m s breasts	0.6000	False Positive
homosexuality scientist discuss, if eugene was homosexual. see therefore: bern ##d - ulrich her ##ge ##mo ##ller , mann fur mann , page 210 magnus hirsch ##feld , die homosexual ##ita ##t des mann ##es und des wei ##bes, 1914 / 1984 , page 66 ##1 albert mo ##ll, be ##ru ##hm ##te homo ##se ##x ##uel ##le, wi ##es ##bad ##en, 1910, page 36	0.6667	False Positive
lo ##l no i meant its a shame its not in common . i have a lot of gay friends so you don ' t need to worry that i am against gay ##s :) in terms of the afc article i can promise i won ##t repeat myself but if i have an advancing argument or need to respond to someone who is clearly wrong then i will do so but in a polite manner only . how about i ask you before i post something to it and if you agree i will ? the problem with this though is that i am up at stupid hours and so you won ' t be up . ok i ' ll look for	0.7238	False Positive
why white people used to say that sex was a sin used to be a mystery to me until i saw the children of browns and mixed race children popping up all around me .	0.1250	False Negative
saudi cleric believes earth does not rev ##ol ##ve around sun . there must be 50 ways to return to the 7th century with # islam as your expert guide	0	False Negative
this ata ##sca ##der ##o woman is accusing of physically abu ##sing two elderly relatives faces felony charges	0.1429	False Negative
rt @ ashes ##cho ##w : women ' s studies : learn how to tell other women they should study something other than women ' s studies rt @ eli ##ah r : http : / / t	0.3750	False Negative
no wonder this races achieve nothing .	0	False Negative
lack of compassion seems to go hand and hand with lack the of white blood.	0	False Negative

Table 5.6: *Examples of false positives assigned with high subjectivity scores and false negatives assigned with low subjectivity scores.*

DATA	RoBERTa				SS-RoBERTa			
	F1	std	False Negative	False Positive	F1	std	False Negative	False Positive
WS	0.5879	0.0233	48	52	**0.5955**	0.0253	46	55
Twitter 18k	0.7975	0.0052	96	151	**0.7992**	0.0072	93	152
Twitter 42k	0.7512	0.0095	70	262	**0.7543**	0.0081	70	256
Wiki	0.7429	0.0077	143	876	**0.7521**	0.0123	126	858

Table 5.7: *Performance comparison between SS-RoBERTa and RoBERTa. The mean F1 score and its standard deviation, the mean false negative and the mean false positive are from 10 independently runs for each model presented. The better F1 result for each task is highlighted in bold.*

on many other NLP tasks (Liu et al., 2019c). While our results may appear inconsistent, some earlier studies reported similar results. For example, Baruah et al. (2020) have shown that BERT (macro F_1 0.6501) outperforms RoBERTa (0.6130)

on the task of aggression identification. Mutanga et al. (2020) have shown that BERT (0.73) outperforms RoBERTa (0.69) in terms of F_1 on the task of hate speech detection. One possible explanation can be that RoBERTa is trained on a much bigger training corpus than BERT. The bigger training corpus of RoBERTa (16GB BERT corpus + 144GB corpus from CC-NEWs, OPENWEBTEXT and CommonCrawl) enhances the knowledge of the formal language. However, content in TCC datasets is written in much more informal language. Another possible reason is that RoBERTa removes the training objective of next sentence prediction from BERT. However, this could have helped the model consider the overall context when interpreting a text.

5.4.6 Limitations

It is unfortunate that the study did not include one of the datasets used by our comparative study due to the dataset availability. The second issue not addressed in this study and will be studied in the future is generalising our method to other pre-trained models which have different structures from BERT, such as Transformer-XL that does not include attention masks (Dai et al., 2019). Third, although the impact of hyperparameters is not the focus of this study, the optimal hyper-parameter setting for each dataset might be different. The potential under-performance due to non-optimal hyper-parameter settings would affect our conclusions. Last, this study is limited to the inaccuracy of subjectivity assigned by the TextBlob. The inaccurate subjectivity scores potentially affect the performance of our methods.

5.5 Conclusion

The identity term bias is commonly found as a limitation of the recent SOTA TCC methods. It affects TCC performance as it often leads to false positive predictions. However, only a few studies have investigated the identity term bias and they tackle the issue based on the same principle of paying less attention to the identity term.

In this chapter, we proposed a novel approach to tackle the identity term bias. This is achieved by training a model to pay additional attention to the subjectivity level of comments only when an identity term appears. Our approach utilises the BERT embeddings structure to embed the information of both subjectivity levels and the presence of identity terms. This has addressed the first research

question for this chapter. In addition to the traditional lexicon-based method, we proposed a new method to measure the subjectivity of a comment when it contains an identity term. The novelty here is that instead of lexicon-based methods in previous works, our method uses the semantic similarity to the relevant Wikipedia summary text of that identity term as a proxy to subjectivity. This has addressed the second research question of measuring the subjectivity level. Our extensive evaluation has answered the third research question: our model SS-BERT outperforms SOTA methods on a wide range of TCC tasks. The results reveal that our method can mitigate the bias toward identity terms and reduce the false positive predictions effectively. Also, the results indicate that semantic similarity calculated by our method potentially reflects the subjectivity level of a comment.

Chapter 6

Conclusion

It is widely acknowledged that creating labelled data for training toxic content classifiers is expensive, while transferring a PLM to a downstream TCC model gets around this issue by allowing the TCC model to utilise a greater amount of unlabelled data via the PLM. Throughout this thesis, we have made contributions to three areas of efficient and effective use of PLMs on TCC tasks: fine-tuning a PLM for TCC tasks (Chapter 3), identifying the identity term bias in PLMs (Chapter 4), and incorporating the subjectivity level to mitigate the identity term bias in PLM-based TCC models (Chapter 5). In this final chapter, we will recapitulate the proposed methods, summarise our findings, and provide an outlook into the future directions.

6.1 Synopsis

In this dissertation, we studied PLM-based methods for automatically detecting toxic comments. Chapter 1 presented the motivation and a preview for this thesis. Chapter 2 reviewed the related concepts and techniques in text classification, particularly deep neural network-based methods. This chapter also introduced transfer learning and pre-trained language models in detail, which are essential foundations for this thesis. Chapter 3 focused on two strategies of transferring PLMs to TCC models: first, how to design the downstream architecture for the PLM; second, how to continue pre-training the PLM in domain with limited computational resources. This chapter also compared different PLMs on TCC tasks and their performances with the two transferring strategies mentioned above. Chapter 4 studied the limitations of SOTA PLMs on TCC tasks. We provided evidences of the identity term bias with model rationales. We also conducted qualitative

and quantitative analysis on false predictions of a simple BERT-based classifier to further investigate the issue of identity term bias and identified the relationship between the toxicity, the subjectivity level and the presence of identity terms. Chapter 5 proposed a novel model built upon BERT to mitigate the identity term bias. The model is intuitively explainable and inspired by the observation and findings from Chapter 4. The model was evaluated on a wide range of TCC tasks and compared to a range of SOTA models. Further, we proved the generalisability of the method with other PLMs built on BERT.

6.2 Summary

6.2.1 Findings

This thesis investigated the techniques of using PLMs for TCC tasks. Over the course of this thesis, we presented extensive comparisons and empirical studies. We also proposed novel methods to improve the performances of current PLMs on TCC tasks. We now recapitulate how our methods addressed the research questions we laid out initially and summarise our findings.

Research question one:

How to configure a PLM-based TCC classifier that achieves optimal performance and efficiency?

There are a range of options for configurations and transferring techniques when applying PLMs to a downstream TCC model. These options are important as it affects what and how the TCC model learns. We addressed this research question by exploring two strategies: the design of downstream neural network architecture and the TAPT-light continued pre-training in domain.

For the first strategy, We conducted extensive experiments to compare different downstream architectures with different PLMs on a wide range of TCC tasks. The results consistently showed that a simple linear downstream architecture performs better than complex ones such as CNN and LSTM. This was an unexpected finding as those complex DNN models usually outperformed the traditional machine learning models when they were used alone without PLMs in previous studies. For the second strategy, we reduced the batch size to examine the TAPT-light performance when using much less computational resources than the original TAPT continued pre-training in domain. We also tested different pre-training epoch values. We found that TAPT-light is beneficial to PLM-based

TCC models, especially on relatively small datasets and the minority classes in the dataset. Regarding the pre-training epoch value, more training epoch does not necessarily bring further improvement. Generally, we recommend 10 or 20 epochs when using TAPT-light, based on the our experiment results. We also compared different PLMs and our results showed that BERT generally outperforms RoBERTa and XLM when they use the identical downstream architecture.

Research question two:

What are the limitations of SOTA PLM-based classifier for TCC?

The current studies have shown promising performance of PLMs in NLP but at the same time, unintended bias in various tasks. The second study of this thesis focused on the limitations of PLMs on TCC tasks, particularly the identity term bias. We addressed this research question by conducting quantitative and qualitative analysis on the BERT predictions on a wide range of TCC tasks. We also examined the model explanations for the erroneous predictions of BERT to study the identity term bias.

Our first finding was that the identity term bias is a prevalent issue among TCC tasks and it was related to the false positive predictions. Second, the identity term bias could be reflected in the rationales of model explanations, which indicate that BERT over-attends to the identity term of non-toxic comments. This potentially contributed to the false positive predictions. Third, false positive predictions generally have a lower subjectivity level than true positive predictions, this implies a potential relationship between the identity term bias and the subjectivity level of comments.

Research question three:

How to handle the identity term bias found in TCC models?

Previous studies usually handled the identity term bias by suppressing the model's attention towards identity terms. We explored a different approach and addressed this research question by taking advantage of the findings from the second study regarding the relationship between the subjectivity level of the comment, the toxicity of the comment and the presence of identity terms. We proposed a novel model, SS-BERT, to incorporate features of the subjectivity level and the presence of identity terms. We also used a method that measures "subjectivity" based on calculating the similarity of a comment to a related Wikipedia page.

Our results led to several findings. First, our model SS-BERT was able to improve PLMs performance by mitigating the identity term bias and decreasing

the false positive predictions. Second, adding the subjectivity level to BERT alone was ineffective as the model did not consistently reduce false positive predictions. This proved our hypothesis that only when the comment contains identity terms, the toxicity of a comment can be associated to the subjectivity level of the comment. Third, the similarity between a comment and the Wikipedia summary for the identity term shared between them potentially indicated the subjectivity level of the comment.

6.2.2 Contributions

We now summarise this thesis' contributions with regard to PLMs for TCC tasks overall and each transfer learning area in particular:

In the first study, we provided a comprehensive comparison of different PLMs, different transferring strategies and different TAPT-light settings on TCC tasks. This empirical study contributed towards enhancing our understanding of applying PLMs to TCC tasks. The findings will also be of interest to future applications of PLMs beyond TCC tasks.

The second study shed new light on the evidence of the identity term bias on TCC tasks via model explanations. The analysis of subjectivity levels undertaken in the study also extended our knowledge of the identity term bias. It identified the relationship between the presence of identity term, the subjectivity level of comments and the toxicity of comments.

The last study proposed a novel method to mitigate the identity term bias. Different from other methods of debiasing the model, our model is based on the intuition that that when someone mentions an identity term, the comment tends to be non-toxic if the comment is objective. To the best of our knowledge, our study was the first one specifically making use of the subjectivity level to handle the identity term bias. This study also showed that the similarity between a comment and the Wikipedia summary regarding the identity term contained by the comment potentially indicates the subjectivity level of the comment.

6.3 Future work

A number of limitations have been identified and future work is proposed. These have been discussed in previous chapters, and are summarised in this section. Additionally, we will also give an outlook into the future for using PLMs on toxic comment classification in general.

6.3.1 On individual studies

TAPT of high computational resource (Chapter 3) Due to our limited computational resources, this study did not evaluate the original TAPT on all TCC tasks. However, we compared their results on one TCC task and TAPT-light achieved a comparable result. It is possible that performances for the other three TCC tasks will be different. Future research might explore the original TAPT further and improve our understanding of the continued pre-training in domain.

Hyperparameters (Chapter 3) We used identical hyper-parameter settings for models with different downstream architectures. This might lead to the under-performance of certain models as some models can have different optimal hyper-parameter settings. To what extent those hyper-parameter settings impact on the performance is unknown. Further research could also be conducted to determine the effectiveness of fine-tuning the hyper-parameters for each model.

New PLMs (Chapter 3) By the time that this study completed, more PLMs are proposed in the latest research, such as TweetBERT and TimeLM which are pre-trained on Twitter corpora and GPT-3 which is designed for few-shot learning (i.e., a learning task with very limited training data) (He et al., 2020; Loureiro et al., 2022; Brown et al., 2020). It is unknown how these new PLMs perform on TCC tasks and whether they will follow the same patterns discovered in this study. A comparative study of these new PLMs could add to our understanding of using PLMs on TCC tasks.

Defining subjectivity levels (Chapter 4) One issue we have found from this study is that among the current TCC studies, there isn't a clear and unified definition for the subjectivity level in the context of TCC or related tasks. This might lead to a debate of our findings and also hinder the application of our method. Therefore, one future research direction is developing a formal definition of subjectivity and methods for quantifying subjectivity of a text. This can benefit a wider range of applications and research that make use of this linguistic feature.

Defining identity terms (Chapter 4) Up to now there is a lack of studies on how to define identity terms or systematically create a list of identity terms. Several questions still remain to be answered. For example, should different TCC tasks use different identity terms? And how to handle identity terms that have many variations (e.g., due to synonyms)? A greater focus on defining identity terms or compiling an identity term list could add values to future TCC research on the identity term bias.

Generalisability of SS-BERT (Chapter 5) One future direction of the third study is to investigate the effect of our method on the other PLMs (Lample and Conneau, 2019; Yang et al., 2019). Although we had examined our method with RoBERTa, experiments with a wider range of PLMs will contribute to understanding the generalisability of our method. Additionally, our method was limited by the dependence of the masking layer. We have not explored how to apply our hypothesis to the PLMs that do not contain a masking layer, such as Transformer-XL (Dai et al., 2019).

Inaccurate subjectivity scores (Chapter 5) It was found that some subjectivity scores by TextBlob were not accurate, which would have misled the model in learning the relationship between subjectivity and toxicity. Further research in SS-BERT may explore other methods of measuring the subjectivity level of comments. Also, we believe more accurate calculation of the subjectivity level can potentially benefit a wide range of tasks.

6.3.2 On the overall research direction

On a broader note, we expect PLMs to be an integral part of TCC models in the coming years. Currently, the usage of PLMs on TCC tasks are limited in several directions. The first is the lack of transparency of the decision making by PLM-based models. Although we have seen the rapid development in both areas of PLMs and TCC methods, there are very limited studies looking at the model behaviour and the model explanation. As mentioned in the survey paper by (Min et al., 2021), "there is a lack of understanding of what actually makes these paradigms (PLMs) so successful, and whether their success can be generalised across models and languages." TCC studies on the model explanation will help answer such questions. Furthermore, the reasoning for the final model prediction could not only provide us insights on improving the model and identifying potential bias, but also contribute to the practical applications. Because the model explanation can be used to assist providing explanations and evidences to the moderation of toxic comments.

Second, due to the limited training data and the high cost of creating labelled datasets, few-shot learning and zero-shot learning with the benefits of PLMs are of great interest for future practical applications. Zero-shot learning is an extreme case of transfer learning, where a model trained on one domain is employed to predict samples from a totally unseen domain (Goodfellow et al., 2016; Zhong et al., 2021; Pamungkas et al., 2021). In few-shot learning, the model is given

only a few examples for each class (Miller et al., 2000; Lake et al., 2011). We notice the growth of such research in recent years, such as work by (Nozza, 2021; Pamungkas et al., 2021; Wang and Banko, 2021). However, these studies diverge in terms of different research objectives, methods and evaluation tasks. Particularly considering the chat-speak style of user generated content from various online platforms such as social media, applying zero- or few-short learning in TCC tasks can be very challenging, and existing research in these areas may not offer much reference value (Nozza, 2021).

6.4 Closing statement

With the explosive growth of social media and discussion forums and the ever-increasing user-generated-content, toxic comments have become a prominent issue online. Although NLP techniques, especially neural networks, have been successfully applied to many practical applications, there is still room for improvement in automated methods for detecting toxic comments online. Our study focused on "transferring" pre-trained language models to TCC and addressing the model bias based on model decision-making explanations. This helps towards building TCC models that are less dependent on training data, and make fairer decisions. Our study also leads to an under-explored direction of using these black-box style deep learning models: manually identifying and applying potential features for a specific task. This idea could be extended to other tasks beyond TCC. Overall, we hope to see that NLP techniques can be used reliably in combating online discrimination and in promoting a healthy and friendly communication environment. We see great potential and bright future of using PLMs on TCC tasks.

Bibliography

Shivang Agarwal and C Ravindranath Chowdary. Combating hate speech using an adaptive ensemble learning model with a case study on covid-19. *Expert Systems with Applications*, 185:115632, 2021.

Charu C Aggarwal. *Data classification: algorithms and applications*. CRC press, 2014.

Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012a.

Charu C Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012b.

Sweta Agrawal and Amit Awekar. Deep learning for detecting cyberbullying across multiple social media platforms. In *European Conference on Information Retrieval*, pages 141–153. Springer, 2018.

Jay Alammar. The illustrated transformer, 2018. URL `shorturl.at/cxOY2`. Last accessed 28 August 2019.

Wafa Alorainy, Pete Burnap, Han Liu, and Matthew L Williams. "the enemy among us": Detecting cyber hate speech with threats-based othering language embeddings. *ACM Transactions on the Web (TWEB)*, 13(3):14, 2019.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

Samir Amin. Eurocentrism, trans. russell moore, 1989.

Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*, 2016.

Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2016.

A Ashwitha, G Shruthi, HR Shruthi, Makarand Upadhyaya, Abhra Pratip Ray, and TC Manjunath. Sarcasm detection in natural language processing. *Materials Today: Proceedings*, 37:3324–3331, 2021.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.263. URL `https://aclanthology.org/2020.emnlp-main.263`.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760. International World Wide Web Conferences Steering Committee, 2017.

Pinkesh Badjatiya, Manish Gupta, and Vasudeva Varma. Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *The World Wide Web Conference*, pages 49–59, 2019.

Shubhanker Banerjee, Bharathi Raja Chakravarthi, and John P McCrae. Comparison of pretrained embeddings to identify hate speech in indian code-mixed text. In *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 21–25. IEEE, 2020.

Nasrin Baratalipour, Ching Y Suen, and Olga Ormandjieva. Abusive language detection using bert pre-trained embedding. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 695–701. Springer, 2020.

Arup Baruah, Kaushik Das, Ferdous Barbhuiya, and Kuntal Dey. Aggression identification in english, hindi and bangla text using bert, roberta and svm. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 76–82, 2020.

Elisa Bassignana, Valerio Basile, and Viviana Patti. Hurtlex: A multilingual lexicon of words to hurt. In *5th Italian Conference on Computational Linguistics, CLiC-it 2018*, volume 2253, pages 1–6. CEUR-WS, 2018.

Matthew Beatty. Graph-based methods to detect hate speech diffusion on twitter. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 502–506. IEEE, 2020.

Fateme Behzadi. Natural language processing and machine learning: A review. *International Journal of Computer Science and Information Security*, 13(9):101, 2015.

Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. A systematic review of reproducibility research in natural language processing. *arXiv preprint arXiv:2103.07929*, 2021.

Emily M Bender and Batya Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 2018.

Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, 2020.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003.

Diego Benito, Oscar Araque, and Carlos A Iglesias. Gsi-upm at semeval-2019 task 5: Semantic similarity and word embeddings for multilingual detection of hate speech against immigrants and women on twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 396–403, 2019.

James Morris Blaut. *The colonizer's model of the world: Geographical diffusionism and Eurocentric history*, volume 1. guilford Press, 1993.

Su Lin Blodgett and Brendan O'Connor. Racial disparity in natural language processing: A case study of social media african-american english. *arXiv preprint arXiv:1707.00061*, 2017.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of "bias" in nlp. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, 2020.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29:4349–4357, 2016.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, and Roger Wattenhofer. On the validity of self-attention as explanation in transformer models. *arXiv preprint arXiv:1908.04211*, 2019.

Michael Buckland and Fredric Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994.

Jacob Buckman. Fair ml tools require problematic ml models. URL `shorturl.at/frGY4`.

Pete Burnap and Matthew L Williams. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data Science*, 5(1):11, 2016.

Pete Burnap, Omer F Rana, Nick Avis, Matthew Williams, William Housley, Adam Edwards, Jeffrey Morgan, and Luke Sloan. Detecting tension in online

communities with computational twitter analysis. *Technological Forecasting and Social Change*, 95:96–108, 2015.

Agostina Calabrese, Michele Bevilacqua, Björn Ross, Rocco Tripodi, and Roberto Navigli. Aaa: Fair evaluation for abuse detection systems wanted. In *13th ACM Web Science Conference 2021*, pages 243–252, 2021.

Jose Camacho-Collados and Mohammad Taher Pilehvar. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788, 2018.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the 2017 ACM on web science conference*, pages 13–22. ACM, 2017.

Vikas S Chavan and SS Shylaja. Machine learning approach for detection of cyber-aggressive comments by peers on social media network. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2354–2358. IEEE, 2015.

Hao Che, Susan McKeever, and Sarah Jane Delany. Abusive text detection using neural networks. 2017.

Hao Chen, Susan Mckeever, and Sarah Jane Delany. Harnessing the power of text mining for the detection of abusive content in social media. In *Advances in Computational Intelligence Systems*, pages 187–205. Springer, 2017.

Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4: 357–370, 2016.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Alexandra Chronopoulou, Christos Baziotis, and Alexandros Potamianos. An embarrassingly simple approach for transfer learning from pretrained language models. *arXiv preprint arXiv:1902.10547*, 2019.

George Chrysostomou and Nikolaos Aletras. Enjoy the salience: Towards better transformer-based faithful explanations with word salience. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8189–8200, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.645. URL `https://aclanthology.org/2021.emnlp-main.645`.

Theodora Chu, Kylie Jue, and Max Wang. Comment abuse classification with deep learning. *Von https://web. stanford. edu/class/cs224n/reports/2762092. pdf abgerufen*, 2016.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *arXiv preprint arXiv:1909.03683*, 2019.

Edward Collins, Nikolai Rozanov, and Bingbing Zhang. Evolutionary data measures: Understanding the difficulty of text classification tasks. *arXiv preprint arXiv:1811.01910*, 2018.

Bullying Prevention Steering Committee. What is cyberbullying, 2019. URL `https://www.stopbullying.gov/cyberbullying/what-is-it/index.html`. Last accessed 19 October 2019.

ConversationAI. Toxic comment classification challenge: Identify and classify toxic online comments, 2017. URL `https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge`.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514, 2021.

Maral Dadvar, FMG de Jong, Roeland Ordelman, and Dolf Trieschnigg. Improved cyberbullying detection using gender information. In *Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop (DIR 2012)*. University of Ghent, 2012.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*, 2017.

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

Oscar Day and Taghi M Khoshgoftaar. A survey on heterogeneous transfer learning. *Journal of Big Data*, 4(1):29, 2017.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, 2018.

Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Technical report, Stanford University, 2008.

Fabio Del Vigna, Andrea Cimino, Felice Dell'Orletta, Marinella Petrocchi, and Maurizio Tesconi. Hate me, hate me not: Hate speech detection on facebook. In *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)*, pages 86–95, 2017.

Li Deng. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3, 2014.

Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, 2020.

Karthik Dinakar, Roi Reichart, and Henry Lieberman. Modeling the detection of textual cyberbullying. In *fifth international AAAI conference on weblogs and social media*, 2011.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73, 2018.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30. ACM, 2015.

Cicero Dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014.

Maeve Duggan. Online harassment: Summary of findings, 2014. URL `https://www.pewinternet.org/2014/10/22/online-harassment/`. Last accessed 22 October 2019.

Mai ElSherief, Shirin Nilizadeh, Dana Nguyen, Giovanni Vigna, and Elizabeth Belding. Peer to peer hate: Hate speech instigators and their targets. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12, 2018.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

Tiziano Fagni, Leonardo Nizzoli, Marinella Petrocchi, and Maurizio Tesconi. Six things i hate about you (in italian) and six classification strategies to more and more effectively find them. In *ITASEC*, 2019.

Yuchun Fang, Zhengyan Ma, Zhaoxiang Zhang, Xu-Yao Zhang, Xiang Bai, et al. Dynamic multi-task learning with convolutional neural network. In *IJCAI*, pages 1668–1674, 2017.

Ibrahim Abu Farha and Walid Magdy. Multitask learning for arabic offensive language and hate-speech detection. In *Proceedings of the 4th workshop on open-source Arabic corpora and processing tools, with a shared task on offensive language detection*, pages 86–90, 2020.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*, 2017.

Anjalie Field, Su Lin Blodgett, Zeerak Waseem, and Yulia Tsvetkov. A survey of race, racism, and anti-racism in nlp. *arXiv preprint arXiv:2106.11410*, 2021.

Benedict Florance. The most intuitive and easiest guide for recurrent neural network, 2018. URL `shorturl.at/gkmoF`. Last accessed 23 August 2019.

Paula Fortuna and Sérgio Nunes. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):85, 2018.

Antigoni Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Twelfth International AAAI Conference on Web and Social Media*, 2018.

Antigoni Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM Conference on Web Science*, pages 105–114. ACM, 2019.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

John Gallacher and Jonathan Bright. Hate contagion: Measuring the spread and trajectory of hate on social media. 2021.

John David Gallacher. *Online Intergroup Conflict: How the dynamics of online communication drive extremism and violence between groups*. PhD thesis, University of Oxford, 2021.

Björn Gambäck and Utpal Kumar Sikdar. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*, pages 85–90, 2017.

Lei Gao and Ruihong Huang. Detecting online hate speech using context aware models. *arXiv preprint arXiv:1710.07395*, 2017.

Zhengjie Gao, Ao Feng, Xinyu Song, and Xi Wu. Target-dependent sentiment classification with bert. *IEEE Access*, 7:154290–154299, 2019.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. Evaluating models' local decision boundaries via contrast sets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1307–1323, 2020.

Ismael Garrido-Muñoz, Arturo Montejo-Ráez, Fernando Martínez-Santiago, and L Alfonso Ureña-López. A survey on bias in deep nlp. *Applied Sciences*, 11(7): 3184, 2021.

Zoubin Ghahramani. Unsupervised learning. In *Summer School on Machine Learning*, pages 72–112. Springer, 2003.

Anthony Gillioz, Jacky Casas, Elena Mugellini, and Omar Abou Khaled. Overview of the transformer-based models for nlp tasks. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 179–183. IEEE, 2020.

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215–230, 2015.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12):2009, 2009.

Yoav Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420, 2016.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Google. Embeddings: Translating to a lower-dimensional space, 2019. URL `shorturl.at/jvwW8`. Last accessed 16 August 2019.

Denis Gordeev and Olga Lykova. Bert of all trades, master of some. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 93–98, 2020.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18 (5-6):602–610, 2005.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. *arXiv preprint arXiv:1803.11138*, 2018.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. A closer look at skip-gram modelling. In *LREC*, pages 1222–1225, 2006.

Matan Halevy, Camille Harris, Amy Bruckman, Diyi Yang, and Ayanna Howard. Mitigating racial biases in toxic language detection with an equity-based ensemble framework. *arXiv preprint arXiv:2109.13137*, 2021.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, et al. Pre-trained models: Past, present and future. *AI Open*, 2021.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. *arXiv preprint arXiv:1610.02413*, 2016.

Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

Maryam Hasan, Emmanuel Agu, and Elke Rundensteiner. Using hashtags as labels for supervised learning of emotions in twitter messages. In *ACM SIGKDD Workshop on Health Informatics, New York, USA*, 2014.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Eric Hilgendorf. The german network enforcement act on the test bench. In *The Quest for Core Values in the Application of Legal Norms*, pages 161–175. Springer, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. *arXiv preprint arXiv:1902.00751*, 2019.

Dirk Hovy and Shannon L Spruit. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–598, 2016.

Ayanna Howard and Jason Borenstein. Trust and bias in robots: These elements of artificial intelligence present ethical challenges, which scientists are trying to solve. *American Scientist*, 107(2):86–90, 2019.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Meiqun Hu, Ee-Peng Lim, Aixin Sun, Hady Wirawan Lauw, and Ba-Quy Vuong. Measuring article quality in wikipedia: models and evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 243–252, 2007.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

Qianjia Huang, Vivek Kumar Singh, and Pradeep Kumar Atrey. Cyber bullying detection using social and textual analysis. In *Proceedings of the 3rd International Workshop on Socially-Aware Multimedia*, pages 3–6. ACM, 2014.

Hairong Huo and Mizuho Iwaihara. Utilizing bert pretrained models with various fine-tune methods for subjectivity detection. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 270–284. Springer, 2020.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL `https://aclanthology.org/2020.acl-main.386`.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C Wallace. Learning to faithfully rationalize by construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4459–4473, 2020.

Soroush Javdan, Behrouz Minaei-Bidgoli, et al. Applying transformers and aspect-based sentiment analysis approaches on sarcasm detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 67–71, 2020.

Shan Jiang and Christo Wilson. Linguistic signals under misinformation and fact-checking: Evidence from user comments on social media. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–23, 2018.

Shan Jiang and Christo Wilson. Structurizing misinformation stories via rationalizing fact-checks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 617–631, 2021.

Jeong Jiwon. The most intuitive and easiest guide for recurrent neural network, 2019. URL `shorturl.at/lnFGI`. Last accessed 16 August 2019.

Vineet John. A survey of neural network techniques for feature extraction from text. *arXiv preprint arXiv:1704.08531*, 2017.

Rie Johnson and Tong Zhang. Supervised and semi-supervised text categorization using lstm for region embeddings. *arXiv preprint arXiv:1602.02373*, 2016.

Prashant Kapil and Asif Ekbal. A deep neural network based multi-task learning approach to hate speech detection. *Knowledge-Based Systems*, 210:106458, 2020.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*, 2019.

Brendan Kennedy, Drew Kogon, Kris Coombs, Joe Hoover, Christina Park, Gwenyth Portillo-Wightman, Aida Mostafazadeh, Mohammad Atari, and

Morteza Dehghani. A typology and coding manual for the study of hate-based rhetoric, 2018.

Brendan Kennedy, Xisen Jin, Aida Mostafazadeh Davani, Morteza Dehghani, and Xiang Ren. Contextualizing hate speech classifiers with post-hoc explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5435–5442, 2020.

Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Pieter-Jan Kindermans, Kristof Schütt, Klaus-Robert Müller, and Sven Dähne. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*, 2016.

Aniket Kittur and Robert E Kraut. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 37–46, 2008.

Dariusz Kłeczek. Polbert: Attacking polish nlp tasks with transformers. In *Proceedings of the PolEval 2020 Workshop*, pages 79–88, 2020.

Vandana Korde and C Namrata Mahender. Text classification and classifiers: A survey. *International Journal of Artificial Intelligence & Applications*, 3(2):85, 2012.

Renard Korzeniowski, Rafał Rolczyński, Przemysław Sadownik, Tomasz Korbak, and Marcin Możejko. Exploiting unsupervised pre-training and automated feature engineering for low-resource hate speech detection in polish. *arXiv preprint arXiv:1906.09325*, 2019.

György Kovács, Pedro Alonso, and Rajkumar Saini. Challenges of hate speech detection in social media. *SN Computer Science*, 2(2):1–15, 2021.

Ritesh Kumar, Aishwarya N. Reganti, Akshit Bhatia, and Tushar Maheshwari. Aggression-annotated Corpus of Hindi-English Code-mixed Data. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan,

May 7-12, 2018 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.

Ritesh Kumar, Atul Kr Ojha, Bornini Lahiri, Marcos Zampieri, Shervin Malmasi, Vanessa Murdock, and Daniel Kadar. Proceedings of the second workshop on trolling, aggression and cyberbullying. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, 2020.

Irene Kwok and Yuzhou Wang. Locate the hate: Detecting tweets against blacks. In *Twenty-seventh AAAI conference on artificial intelligence*, 2013.

Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.

Eric Lavergne, Rajkumar Saini, György Kovács, and Killian Murphy. Thenorth@ haspeede 2: Bert-based language model fine-tuning for italian hate speech detection. In *Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020), Online, Decembert 17, 2020*. RWTH Aachen University, 2020.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases. In *International Conference on Machine Learning*, pages 1078–1088. PMLR, 2020.

Hana Lee and Young Yoon. Engineering doc2vec for automatic classification of product descriptions on o2o applications. *Electronic Commerce Research*, 18(3): 433–456, 2018.

Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 121–126, 2017.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

Nayeon Lee, Belinda Z Li, Sinong Wang, Pascale Fung, Hao Ma, Wen-tau Yih, and Madian Khabsa. On unifying misinformation detection. *arXiv preprint arXiv:2104.05243*, 2021.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, 2016.

Derczynski Leon, Vidgen Bertie, Kirk Hannah, Rose, Johansson Pica, Chung Yi-Ling, Kjeldgaard Kongsbak Mads, Guldborg, Sprejer Laila, and Zeinert Philine. hatespeechdata: Hate speech dataset catalogue. URL `https://hatespeechdata.com/`.

Edwin Efraín Jiménez Lepe and Andrés Méndez Vázquez. A beginner's tutorial for cnn. 2017.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Chenghua Lin, Yulan He, and Richard Everson. Sentence subjectivity detection with weakly-supervised learning. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1153–1161, 2011.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016.

Michael Lissack. The slodderwetenschap (sloppy science) of stochastic parrots — a plea for science to not take the route advocated by gebru and bender, 2021. URL `shorturl.at/atBRS`. Last accessed 01 Jan 2022.

Han Liu, Pete Burnap, Wafa Alorainy, and Matthew L Williams. Fuzzy multi-task learning for hate speech type identification. In *The World Wide Web Conference*, pages 3006–3012. ACM, 2019a.

Han Liu, Peter Burnap, Wafa Alorainy, and Matthew Williams. Scmhl5 at trac-2 shared task on aggression identification: Bert based ensemble learning approach. 2020.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019b.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. 2019c.

Rachel Tsz-Wai Lo, Ben He, and Iadh Ounis. Automatically building a stopword list for an information retrieval system. In *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR)*, volume 5, pages 17–24, 2005.

Steven Loria. textblob documentation. *Release 0.15*, 2, 2018.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-Collados. Timelms: Diachronic language models from twitter. *arXiv preprint arXiv:2202.03829*, 2022.

Jie Lu, Vahid Behbood, Peng Hao, Hua Zuo, Shan Xue, and Guangquan Zhang. Transfer learning using computational intelligence: a survey. *Knowledge-Based Systems*, 80:14–23, 2015.

Denis Lukovnikov, Asja Fischer, and Jens Lehmann. Pretrained transformers for simple question answering over knowledge graphs. In *International Semantic Web Conference*, pages 470–486. Springer, 2019.

Alessandro Lusci, Gianluca Pollastri, and Pierre Baldi. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-

like molecules. *Journal of chemical information and modeling*, 53(7):1563–1575, 2013.

Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. Hate speech detection: Challenges and solutions. *PloS one*, 14(8):e0221152, 2019.

Raghvendra Mall, Mridul Nagpal, Joni Salminen, Hind Almerekhi, Soon-Gyo Jung, and Bernard J Jansen. Four types of toxic people: characterizing online users' toxicity over time. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*, pages 1–11, 2020.

Shervin Malmasi and Marcos Zampieri. Challenges in discriminating profanity from hate speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2):187–202, 2018.

Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de La Clergerie, Djamé Seddah, and Benoît Sagot. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*, 2019.

Masakazu Matsugu, Katsuhiko Mori, Yusuke Mitari, and Yuji Kaneda. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16(5-6):555–559, 2003.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305, 2017.

Priscilla Marie Meddaugh and Jack Kay. Hate speech or "reasonable racism?" the other in stormfront. *Journal of Mass Media Ethics*, 24(4):251–268, 2009.

Yashar Mehdad and Joel Tetreault. Do characters abuse more than words? In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 299–303, 2016.

Mostafa Mesgari, Chitu Okoli, Mohamad Mehdi, Finn Årup Nielsen, and Arto Lanamäki. "the sum of all human knowledge": A systematic review of scholarly

research on the content of w ikipedia. *Journal of the Association for Information Science and Technology*, 66(2):219–245, 2015.

Tomas Mikolov, Jiri Kopecky, Lukas Burget, Ondrej Glembek, et al. Neural network based language models for highly inflective languages. In *2009 IEEE international conference on acoustics, speech and signal processing*, pages 4725–4728. IEEE, 2009.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Erik G Miller, Nicholas E Matsakis, and Paul A Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471. IEEE, 2000.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv preprint arXiv:2111.01243*, 2021.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. A bert-based transfer learning approach for hate speech detection in online social media. In *International Conference on Complex Networks and Their Applications*, pages 928–940. Springer, 2019.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. Hate speech detection and racial bias mitigation in social media based on bert model. *PloS one*, 15 (8):e0237861, 2020.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1, pages 1–5. IEEE, 2019.

R Mutanga, Nalindren Naicker, and Oludayo O Olugbara. Hate speech detection in twitter using transformer methods. *International Journal of Advanced Computer Science and Applications*, 11(01), 2020.

Vinita Nahar, Sanad Al-Maskari, Xue Li, and Chaoyi Pang. Semi-supervised learning for cyberbullying detection in social networks. In *Australasian Database Conference*, pages 160–171. Springer, 2014.

Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *Transactions on Asian and Low-Resource Language Information Processing*, 20(5):1–35, 2021.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee, 2016.

Debora Nozza. Exposing the limits of zero-shot cross-lingual hate speech detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 907–914, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.114. URL `https://aclanthology.org/2021.acl-short.114`.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

Atte Oksanen, James Hawdon, Emma Holkeri, Matti Näsi, and Pekka Räsänen. Exposure to online hate among young social media users. *Sociological studies of children & youth*, 18(1):253–273, 2014.

Francisco Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1): 115, 2016.

Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. Multilingual and multi-aspect hate speech analysis. In *EMNLP/IJCNLP (1)*, 2019.

Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. *Information Processing & Management*, 58(4):102544, 2021.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, 2004.

Ji Ho Park and Pascale Fung. One-step and two-step classification for abusive language detection on twitter. In *Proceedings of the First Workshop on Abusive Language Online*, pages 41–45, 2017.

Ji Ho Park, Jamin Shin, and Pascale Fung. Reducing gender bias in abusive language detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2799–2804, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `shorturl.at/opMR0`.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. Deep learning for user comment moderation. *arXiv preprint arXiv:1705.09993*, 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018a.

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

Quang Huu Pham, Viet Anh Nguyen, Linh Bao Doan, Ngoc N Tran, and Ta Minh Thanh. From universal language model to downstream task: Improving roberta-

based vietnamese hate speech detection. In *2020 12th International Conference on Knowledge and Systems Engineering (KSE)*, pages 37–42. IEEE, 2020.

Flor Miriam Plaza-del Arco, M Dolores Molina-González, L Alfonso Ureña-López, and M Teresa Martín-Valdivia. Comparing pre-trained language models for spanish hate speech detection. *Expert Systems with Applications*, 166:114120, 2021.

David Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2 (1):37–63, 2011.

Purvi Prajapati, Amit Thakkar, and Amit Ganatra. A survey and current research challenges in multi-label classification methods. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(1):248–252, 2012.

Flavien Prost, Hai Qian, Qiuwen Chen, Ed H Chi, Jilin Chen, and Alex Beutel. Toward a better trade-off between performance and fairness with kernel-based distribution matching. *arXiv preprint arXiv:1910.11779*, 2019.

Matthew Purver and Stuart Battersby. Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–491. Association for Computational Linguistics, 2012.

Jing Qian, Mai ElSherief, Elizabeth Belding, and William Yang Wang. Hierarchical cvae for fine-grained hate speech classification. *arXiv preprint arXiv:1809.00088*, 2018.

Mohiuddin Md Abdul Qudar and Vijay Mago. Tweetbert: A pretrained language representation model for twitter text analysis. *arXiv preprint arXiv:2010.11091*, 2020.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

Md Mustafizur Rahman, Dinesh Balakrishnan, Dhiraj Murthy, Mucahid Kutlu, and Matthew Lease. An information retrieval approach to building datasets for hate speech detection. *arXiv preprint arXiv:2106.09775*, 2021.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.

Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

Amir H Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. Offensive language detection using multi-level classification. In *Canadian Conference on Artificial Intelligence*, pages 16–27. Springer, 2010.

Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgilio AF Almeida, and Wagner Meira Jr. " like sheep among wolves": Characterizing hateful users on twitter. *arXiv preprint arXiv:1801.00317*, 2017.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, 2020.

Julian Risch, Eva Krebs, Alexander Löser, Alexander Riese, and Ralf Krestel. Fine-grained classification of offensive language. In *14th Conference on Natural Language Processing KONVENS 2018*, page 38, 2018.

Julian Risch, Anke Stoll, Marc Ziegele, and Ralf Krestel. hpidedis at germeval 2019: Offensive language identification using a german bert model. In *KONVENS*, 2019.

Campbell Robertson, Christopher Mele, and Sabrina Tavernise. 11 killed in synagogue massacre; suspect charged with 29 counts. *The New York Times*, Oct 2018. URL `shorturl.at/fyIKR`.

James A Rodger and Parag C Pendharkar. A field study of the impact of gender and user's technical experience on the performance of voice-activated medical tracking application. *International Journal of Human-Computer Studies*, 60 (5-6):529–544, 2004.

Anna Rogers. Changing the world by changing the data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2182–2194, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.170. URL `https://aclanthology.org/2021.acl-long.170`.

Anna Rogers, Timothy Baldwin, and Kobi Leins. 'just what do you think you're doing, dave?'a checklist for responsible data use in nlp. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4821–4833, 2021.

Sebastian Ruder. *Neural Transfer Learning for Natural Language Processing*. PhD thesis, NATIONAL UNIVERSITY OF IRELAND, GALWAY, 2019.

Sebastian Ruder12, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Sluice networks: Learning what to share between loosely related tasks. *stat*, 1050:23, 2017.

Semiu Salawu, Yulan He, and Joanna Lumsden. Approaches to automated detection of cyberbullying: A survey. *IEEE Transactions on Affective Computing*, 2017.

Haji Mohammad Saleem, Kelly P Dillon, Susan Benesch, and Derek Ruths. A web of hate: Tackling hateful speech in online social spaces. *arXiv preprint arXiv:1709.10159*, 2017.

Nithya Sambasivan, Erin Arnesen, Ben Hutchinson, Tulsee Doshi, and Vinodkumar Prabhakaran. Re-imagining algorithmic fairness in india and beyond. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 315–328, 2021.

Niloofar Safi Samghabadi, Parth Patwa, PYKL Srinivas, Prerana Mukherjee, Amitava Das, and Thamar Solorio. Aggression and misogyny detection using bert: A multi-task approach. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 126–131, 2020.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Yogesh Sankarasubramaniam, Krishnan Ramanathan, and Subhankar Ghosh. Text summarization using wikipedia. *Information Processing & Management*, 50(3):443–461, 2014.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. The risk of racial bias in hate speech detection. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1668–1678, 2019.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Anna Schmidt and Michael Wiegand. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, 2017.

Daniel Thilo Schroeder, Ferdinand Schaal, Petra Filkukova, Konstantin Pogorelov, and Johannes Langguth. Wico graph: A labeled dataset of twitter subgraphs based on conspiracy theory and 5g-corona misinformation tweets. In *ICAART (2)*, pages 257–266, 2021.

Fabrizio Sebastiani and Andrea Esuli. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 417–422, 2006.

Sofia Serrano and Noah A Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, 2019.

Nadia Felix F Da Silva, Luiz FS Coletta, and Eduardo R Hruschka. A survey and comparative study of tweet sentiment analysis via semi-supervised learning. *ACM Computing Surveys (CSUR)*, 49(1):15, 2016.

Mohammad S Sorower. A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, 18:1–25, 2010.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

Gary Stein, Bing Chen, Annie S Wu, and Kien A Hua. Decision tree classifier for network intrusion detection with ga-based feature selection. In *Proceedings of the 43rd annual Southeast regional conference-Volume 2*, pages 136–141. ACM, 2005.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Jared Suttles and Nancy Ide. Distant supervision for emotion classification with discrete binary values. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 121–136. Springer, 2013.

Kaveh Taghipour and Hwee Tou Ng. Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 314–323, 2015.

Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer, 2018.

Yi Chern Tan and L Elisa Celis. Assessing social and intersectional biases in contextualized word representations. *arXiv preprint arXiv:1911.01485*, 2019.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565, 2014.

Matthew Tang, Priyanka Gandhi, Md Ahsanul Kabir, Christopher Zou, Jordyn Blakey, and Xiao Luo. Progress notes classification and keyword extraction using attention-based deep learning models with bert. *arXiv preprint arXiv:1910.05786*, 2019.

Rachael Tatman. Gender and dialect bias in youtube's automatic captions. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 53–59, 2017.

Sabrina The New York Times. Christchurch shooting live updates: 49 are dead after 2 mosques are hit. *The New York Times*, March 2019. URL `shorturl.at/oEGN5`.

Songül Tolan, Marius Miron, Emilia Gómez, and Carlos Castillo. Why machine learning may lead to unfairness: Evidence from risk assessment for juvenile justice in catalonia. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pages 83–92, 2019.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov): 45–66, 2001.

Ibtissam Touahri and Azzeddine Mazroui. Enhancement of a multi-dialectal sentiment analysis system by the detection of the implied sarcastic features. *Knowledge-Based Systems*, page 107232, 2021.

Ana-Sabina Uban and Liviu P Dinu. On transfer learning for detecting abusive language online. In *International Work-Conference on Artificial Neural Networks*, pages 688–700. Springer, 2019.

Stefanie Ullmann and Marcus Tomalin. Quarantining online hate speech: technical and ethical perspectives. *Ethics and Information Technology*, 22(1):69–80, 2020.

Elise Fehn Unsvåg and Björn Gambäck. The effects of user features on twitter hate speech detection. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 75–85, 2018.

Alper Kursat Uysal. An improved global feature selection scheme for text classification. *Expert systems with Applications*, 43:82–92, 2016.

Alper Kursat Uysal and Serkan Gunal. The impact of preprocessing on text classification. *Information Processing & Management*, 50(1):104–112, 2014.

Ameya Vaidya, Feng Mai, and Yue Ning. Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 683–693, 2020.

Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 33–42, 2018.

Cynthia Van Hee, Els Lefever, Ben Verhoeven, Julie Mennes, Bart Desmet, Guy De Pauw, Walter Daelemans, and Véronique Hoste. Detection and fine-grained classification of cyberbullying events. In *Proceedings of the international conference recent advances in natural language processing*, pages 672–680, 2015.

Cynthia Van Hee, Gilles Jacobs, Chris Emmery, Bart Desmet, Els Lefever, Ben Verhoeven, Guy De Pauw, Walter Daelemans, and Véronique Hoste. Automatic detection of cyberbullying in social media text. *PloS one*, 13(10):e0203794, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761*, 2020.

Immanuel Wallerstein. Eurocentrism and its avatars: The dilemmas of social science. *Sociological bulletin*, 46(1):21–39, 1997.

Cindy Wang and Michele Banko. Practical transformer-based multilingual text classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 121–129, 2021.

Jin Wang, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. Dimensional sentiment analysis using a regional cnn-lstm model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 225–230, 2016a.

Tinghua Wang, Junyang Rao, and Qi Hu. Supervised word sense disambiguation using semantic diffusion kernel. *Engineering Applications of Artificial Intelligence*, 27:167–174, 2014.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, 2016b.

Zeerak Waseem. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142, 2016a.

Zeerak Waseem. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 138–142, Austin, Texas, November 2016b. Association for Computational Linguistics. URL `http://aclweb.org/anthology/W16-5618`.

Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93, 2016.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. Understanding abuse: A typology of abusive language detection subtasks. *arXiv preprint arXiv:1705.09899*, 2017.

Zeerak Waseem, James Thorne, and Joachim Bingel. Bridging the gaps: Multi task learning for domain transfer of hate speech detection. In *Online Harassment*, pages 29–55. Springer, 2018.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016.

Aksel Ladegård Wester, Lilja Øvrelid, Erik Velldal, and Hugo Lewi Hammer. Threat detection in online discussions. 2016.

Daniel Westreich, Justin Lessler, and Michele Jonsson Funk. Propensity score estimation: neural networks, support vector machines, decision trees (cart), and meta-classifiers as alternatives to logistic regression. *Journal of clinical epidemiology*, 63(8):826–833, 2010.

Gregor Wiedemann, Eugen Ruppert, Raghav Jindal, and Chris Biemann. Transfer learning from lda to bilstm-cnn for offensive language detection in twitter. *arXiv preprint arXiv:1811.02906*, 2018.

Michael Wiegand, Josef Ruppenhofer, and Thomas Kleinbauer. Detection of abusive language: the problem of biased datasets. 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 602–608, 2019.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002. URL `https://aclanthology.org/D19-1002`.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*, 2015.

Ruth Wodak and Martin Reisigl. Discourse and racism: European perspectives. *Annual review of anthropology*, 28(1):175–199, 1999.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

Zhengxuan Wu and Desmond C Ong. On explaining your explanations of bert: An empirical study with sequence classification. *arXiv preprint arXiv:2101.00196*, 2021.

Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. Demoting racial bias in hate speech detection. *arXiv preprint arXiv:2005.12246*, 2020.

Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1980–1984. ACM, 2012.

Albert Xu, Eshaan Pathak, Eric Wallace, Suchin Gururangan, Maarten Sap, and Dan Klein. Detoxifying language models risks marginalizing minority voices. *arXiv preprint arXiv:2104.06390*, 2021.

Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4):611–629, 2018.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

Shiren Ye, Tat-Seng Chua, and Jie Lu. Summarizing definition from wikipedia. In *Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP*, pages 199–207, 2009.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018a.

Haiyang Zhang, Alison Sneyd, and Mark Stevenson. Robustness and reliability of gender bias assessment in word embeddings: The role of base pairs. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 759–769, 2020.

Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018b.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.

Wei Zhang and Clement Yu. Uic at trec 2006 blog track: a notebook paper. *In Proc. of TREC 2006*, 2006.

Wei Zhang, Clement Yu, and Weiyi Meng. Opinion retrieval from blogs. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 831–840, 2007.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

Ziqi Zhang and Lei Luo. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, 10(5):925–945, 2019.

Ziqi Zhang, David Robinson, and Jonathan Tepper. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference*, pages 745–760. Springer, 2018c.

Zhixue Zhao, Ziqi Zhang, and Frank Hopfgartner. Detecting toxic content online and the effect of training data on classification performance. Technical report, EasyChair, 2019.

Zhixue Zhao, Ziqi Zhang, and Frank Hopfgartner. A comparative study of using pre-trained language models for toxic comment classification. In *Companion Proceedings of the Web Conference 2021*, pages 500–507, 2021.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. *arXiv preprint arXiv:2104.04670*, 2021.

Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. Challenges in automated debiasing for toxic language detection. *arXiv preprint arXiv:2102.00086*, 2021.

Appendices

Zhixue Zhao
Registration number: 180275223
Information School
Programme: INFR31 Information Studies (PhD/Info Studs (SSc) FT)

Dear Zhixue

PROJECT TITLE: Handling limited data with transfer learning in toxic comment classification tasks.
APPLICATION: Reference Number 031591

On behalf of the University ethics reviewers who reviewed your project, I am pleased to inform you that on 08/12/2019 the above-named project was **approved** on ethics grounds, on the basis that you will adhere to the following documentation that you submitted for ethics review:

- University research ethics application form 031591 (form submission date: 07/11/2019); (expected project end date: 31/12/2022).

If during the course of the project you need to deviate significantly from the above-approved documentation please inform me since written approval will be required.

Your responsibilities in delivering this research project are set out at the end of this letter.

Yours sincerely

Paul Reilly
Ethics Administrator
Information School

Please note the following responsibilities of the researcher in delivering the research project:

- The project must abide by the University's Research Ethics Policy: https://www.sheffield.ac.uk/rs/ethicsandintegrity/ethicspolicy/approval-procedure
- The project must abide by the University's Good Research & Innovation Practices Policy: https://www.sheffield.ac.uk/polopoly_fs/1.671066!/file/GRIPPolicy.pdf
- The researcher must inform their supervisor (in the case of a student) or Ethics Administrator (in the case of a member of staff) of any significant changes to the project or the approved documentation.
- The researcher must comply with the requirements of the law and relevant guidelines relating to security and confidentiality of personal data.
- The researcher is responsible for effectively managing the data collected both during and after the end of the project in line with best practice, and any relevant legislative, regulatory or contractual requirements.

Application 031591

Section A: Applicant details

Date application started:
Mon 4 November 2019 at 00:28

First name:
Zhixue

Last name:
Zhao

Email:
zhixue.zhao@sheffield.ac.uk

Programme name:
INFR31 Information Studies (PhD/Info Studs (SSc) FT)

Module name:
Standard PhD Thesis
Last updated:
08/12/2019

Department:
Information School

Applying as:
Postgraduate research

Research project title:
Handling limited data with transfer learning in toxic comment classification tasks.

Has your research project undergone academic review, in accordance with the appropriate process?
Yes

Similar applications:
020777 ''Multi-Label Classification for Toxic Comments Detection at Scale with Machine Learning''

Section B: Basic information

Supervisor

Name	Email
Ziqi Zhang	ziqi.zhang@sheffield.ac.uk

Proposed project duration

Start date (of data collection):
Sat 30 November 2019

Anticipated end date (of project)
Sat 31 December 2022

3: Project code (where applicable)

Project externally funded?
- not entered -

Project code
- not entered -

Suitability

Takes place outside UK?
No

Involves NHS?
No

Health and/or social care human-interventional study?
No

ESRC funded?
No

Likely to lead to publication in a peer-reviewed journal?
Yes

Led by another UK institution?
No

Involves human tissue?
No

Clinical trial or a medical device study?
No

Involves social care services provided by a local authority?
No

Is social care research requiring review via the University Research Ethics Procedure
No

Involves adults who lack the capacity to consent?
No

Involves research on groups that are on the Home Office list of 'Proscribed terrorist groups or organisations?
No

Indicators of risk

Involves potentially vulnerable participants?
No
Involves potentially highly sensitive topics?
Yes

Section C: Summary of research

1. Aims & Objectives

The research aims to extend the understanding of using transfer learning to handle limited data in toxic comment classification tasks, especially from three perspectives (i.e. three research questions):
a). Discover the most beneficial existing pre-trained model (i.e. transfer learning methods) for Toxic Comment Classification (TCC) tasks, b). Explore the transfer learning model pre-trained on user classification for TCC, and c). Explore the hybrid model combining transfer learning and multi-task learning for TCC tasks.

Based on the aims, the four objectives of the research are as follows:
1. To review related literature and study the knowledge of text classification, supervised machine learning, transfer learning and multi-task learning.
2. To build different transfer learning models and test them on a wide range of TCC tasks, giving a comprehensive comparison of transfer learning methods.
3. To develop a transfer learning model pre-trained on user classification tasks and adapt it to TCC tasks, investigating the effect of this pre-trained model on TCC tasks.
4. To develop a hybrid model which generally utilising more data and to investigate its impact on TCC tasks.

2. Methodology

This research mainly uses quantitative research methods, which compare different machine learning techniques and models on the tasks of toxic comment classification. It is essentially a comparative study. In total, we have three studies, and each of them will address a research question described in the ''Aims & Objectives''. In each study, we will build several different prediction models and one baseline model. All models will be evaluated on publicly available datasets to investigate their predictive performance on a wide range of tasks. Note that this research uses ''model'' to only refer to the machine learning models in natural language processing applications.

To be more specific:
- In study one, to investigate the performance of different transfer learning methods on Toxic Comment Classification (TCC), we will compare existing publicly available pre-trained models on a wide range of TCC tasks. The pre-trained models are available to download directly, which essentially are codes of the model with parameters learnt already. TCC tasks are publicly available datasets, published by previous researchers or research organisations. They are also known as secondary data. We describe each of the datasets in detail and their publications information in the later section of this application. For some datasets, we need to use Twitter APIs to gather some data since the published datasets do not contain the text directly. For example, dataset D4, as we specified in the additional document section, contains the tweets IDs and labels for each instance. In such a situation, we need to use the Twitter API to collect the tweet content by querying with the tweet IDs.

- In study two, we hypothesise that the user classification task model can help with the TCC tasks by transfer learning. To test this hypothesis, we will build a transfer learning model which transfers the user classification model to TCC tasks. To do so, we will pre-train a model on user classification tasks first using user classification datasets and then transfer it (the model and its parameters) to the TCC tasks. The TCC tasks are the same as mentioned in study one above. The user classification tasks are also publicly available published by previous researchers (we list two of them in section F in this application form).

- In study three, we propose to build a hybrid model which combines transfer learning and multi-task learning, two machine learning paradigm, in one model. We will build the model first and then test it on a wide range of TCC tasks. The TCC tasks will use the same datasets of the ones in study one and study two as described above.

Analysis:
Our three studies in this research are essentially comparative studies. In short, we will compare the proposed model with each other, and also with the baseline model on the same tasks (TCC tasks) using a unified measure metrics. By a wide range of comparison, we will be able to analyse each model and its advantages and disadvantages.

This research will mainly use quantitative research methods and thus will only report aggregated results rather than any original text. We may conduct some qualitative analysis to understand the errors made in the model. In this situation, we will not publish any specific and complete textual content. We may use a snippet from one instance as an example to discuss our findings. Any specific and related user information will not be published (any identifiable information already been removed from the dataset). For example, if we want to consider the effect of typos on models, we will mention the typo word rather than the whole sentence or text.

3. Personal Safety

Have you completed your departmental risk assessment procedures, if appropriate?

Not applicable

Raises personal safety issues?

No

All the works of this research will be implemented using Python programming in computer Labs. No foreseeable physical risk. In terms of mental well-being, this research uses secondary data and mainly assesses the model performance rather than the textual content to be classified itself. Therefore, researchers will barely be exposed to content which potentially impacts researchers' mental well-being.

Section D: About the participants

1. Potential Participants

This research uses secondary dataset which does not require to recruit participants. Some datasets only released the tweet IDs (as shown by example B) rather than the real content (as shown by example A). In these cases, we will retrieve the original tweet text to match the original dataset. Therefore, indirect participants are those that created these content or those that are involved in the dataset. In any situation, any identifiable information of users will not be collected.

Example A:
Instances of TCC dataset:
normal; "I dont care how you think"
aggressive; "go kill yourself"

Example B:
Instances of TCC dataset (contains tweet IDs instead of tweet content):

normal, 683189274167095297
aggressive, 683134555264528385

2. Recruiting Potential Participants

As described above, this research uses secondary dataset, which does not require to recruit participants. As explained above, we will use the Twitter API to collect tweets by using tweet IDs provided in the secondary dataset.

2.1. Advertising methods

Will the study be advertised using the volunteer lists for staff or students maintained by CiCS? No

- not entered -

3. Consent

Will informed consent be obtained from the participants? (i.e. the proposed process) No

Firstly, according to the Twitter privacy policy (https://twitter.com/en/privacy), Twitter users that choose to make their profile public have accepted that their data will be publicly available via the Twitter API to research uses.

Secondly, in legal terms, this does not breach the GDPR clause about "Lawfulness of processing". Article 6, 1 identifies six situations under which collecting and processing personal data is lawful. Our research can be classified under 1(f) 'processing is necessary for the legitimate interests', as mentioned in Rectal 47 (http://www.privacy-regulation.eu/en/r47.htm). As set out above in the Twitter policy, users who share their data publicly should reasonably expect that their public data can be used for research purposes by universities. Thus we argue this research represents 'legitimate interest'.

Thirdly, the scale of the data collection also means that obtaining explicit consent from tens of thousands of users is not applicable. Some data only contains the text without any user information which makes it impossible to reach the participants. The data contains account IDs or screen-names will be represented by random digits rather than meaningful symbols.

Finally, in reality, users often choose not to read privacy policy entirely and as a result, may not understand that their data could be used for research purpose. We will mitigate this risk by ensuring user identities are properly anonymised and that it is not possible to re-identify users from the disseminated work. Specifically:
- We will make sure not to develop methods to identify individual users, but toxic comment classifier tools. The user metadata which is anonymised and represented by abstract numbers is insufficient to re-identify individual users.
- We delete any user's screennames and replace them with random and meaningless numbers.
- We will only publish the performance of models. We will make sure not to release any individual user information in any form. Only the aggregated data, such as the demographic population distribution.
- We will follow ethical rules strictly and not share any data outside the research,

4. Payment

Will financial/in kind payments be offered to participants? No

5. Potential Harm to Participants

What is the potential for physical and/or psychological harm/distress to the participants?

There is no potential harm for the participants since they can not be identified via the data to be processed in this research.

How will this be managed to ensure appropriate protection and well-being of the participants?

We will make sure the data is fully anonymised.

Section E: About the data

1. Data Processing

Will you be processing (i.e. collecting, recording, storing, or otherwise using) personal data as part of this project? (Personal data is any information relating to an identified or identifiable living person).
Yes

Which organisation(s) will act as Data Controller?

University of Sheffield only

2. Legal basis for processing of personal data

The University considers that for the vast majority of research, 'a task in the public interest' (6(1)(e)) will be the most appropriate legal basis. If, following discussion with the UREC, you wish to use an alternative legal basis, please provide details of the legal basis, and the reasons for applying it, below:

- not entered -

Will you be processing (i.e. collecting, recording, storing, or otherwise using) 'Special Category' personal data?
Yes

The University considers the most appropriate condition to be that 'processing is necessary for archiving purposes in the public interest, scientific research purposes or statistical purposes' (9(2)(j)) . If, following discussion with the UREC, you wish to use an alternative condition, please provide details of the condition, and the reasons for applying it, below:

- not entered -

3. Data Confidentiality

What measures will be put in place to ensure confidentiality of personal data, where appropriate?

1. The data collection is only conducted on the University server machines. Once data collection is complete they will be anonymised and then encrypted and stored in only University drive.
2. All following processing and experiments will only be conducted using university computers.
3. Identifiable information will be discarded.

4. Data Storage and Security

In general terms, who will have access to the data generated at each stage of the research, and in what form

I and my supervisors, Dr Ziqi Zhang and Dr Frank Hopfgartner, will have access to the data which are anonymised.

What steps will be taken to ensure the security of data processed during the project, including any identifiable personal data, other than those already described earlier in this form?

We will only use the university computer to process and store data. Also, we do not have any identifiable personal data.

Will all identifiable personal data be destroyed once the project has ended?
Yes

Please outline when this will take place (this should take into account regulatory and funder requirements).

All datasets will be anonymised once it is collected. Any identifiable personal data are destroyed within the project.

Section F: Supporting documentation

Information & Consent

Participant information sheets relevant to project?
No

Consent forms relevant to project?
No

Additional Documentation

Document 1072302 (Version 1) — All versions
Summary of datasets and their source links

External Documentation

The detailed information of each dataset is listed in the additional documentation. The source of each dataset is as below:

Eight publicly available datasets (twitter dataset two and three are from the same source):
D1 -
Stormfront dataset: https://www.aclweb.org/anthology/W18-5102.pdf
D2 -
Facebook dataset: https://www.aclweb.org/anthology/W18-4401/
D3 -
Toxic Comment Classification Challenge (Kaggle competition, Wikipedia dataset): https://www.kaggle.com/c/jigsaw-toxiccomment-classification-challenge
D4 -
Detecting Insults in Social Commentary (Kaggle competition dataset): https://www.kaggle.com/c/detecting-insults-in-social-commentary
D5 & D6 -
Twitter dataset one and two: https://github.com/zeerakw/hatespeech
D7 -

Twitter dataset three: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/download/15665/14843
D8 -
Twitter dataset four: http://www.yichang-cs.com/yahoo/WWW16_Abusivedetection.pdf

Example instances of TCC dataset:
normal; "I dont care how you think"
aggressive; "go kill yourself"

Example instances of TCC dataset (contains tweet IDs instead of tweet content):
normal, 683189274167095297
aggressive, 683134555264528385

Two user classification tasks:
D9 -
Hateful Users on Twitter(Kaggle competition): https://www.kaggle.com/manoelribeiro/hateful-users-on-twitter
D10 -
Twitter mean birds: https://arxiv.org/abs/1702.06877

Example instances of user classification datasets (D10 format, which does not contain tweet contents but tweet IDs):
user_1 aggressor
657684957272387584,657695791134416896,656838718666493952,657193317709565952,657688224601632768,657771833853411328,657231280367673344,657253568878157824,6569525311C
user_2 aggressor 663286416676274176,663345753515945984,663271094904754176,663301981742592000,663304166995968000,663300853479018496,663297196054347776
user_3 aggressor 683165798660206592,683189274167095297,683134555264528385,683110668699086849,683192428304662528
user_4 aggressor 737427866032295936,737284416603492356,737327198533623808,737171166230511616,737398925301145602,737503355111542785

Section G: Declaration

Signed by:
ZHIXUE ZHAO
Date signed:
Thu 7 November 2019 at 00:36

Offical notes

- not entered -