Final Year Project

Online Hate Detection

Kier McGuirk

Student ID: 18752609

A thesis submitted in part fulfilment of the degree of

BSc. (Hons.) in Computer Science

Supervisor: Professor Simon Caton

UCD School of Computer Science

University College Dublin

May 6, 2022

Contents

Note: The GitLab repository containing the project code, primary datasets and models is accessible at:
https://gitlab.com/kiermcguirk/online-hate-detection/

Abstract

The epidemic of *"Online Hate"* is prolific in the online realm, posing severe psychological consequences to victims and has become a problem often considered as, if not more, exigent than physical bullying. This epidemic concomitantly invented the field of Online Hate Detection within natural language processing – a notion predicated on utilising classification techniques to identify hateful text in online communities. Many forms of *"Online Hate Detectors"* have been implemented to combat it with remarkable results, and over the years, the approaches have become more robust and effective. However, Online Hate Detectors tailored for use in Ireland is a sparse and under-researched topic. This research synthesised a modern and general Online Hate Detection mechanism tailored for use in Ireland. Twitter and Wikipedia data were collated and labelled either 'hateful' or 'not hateful' to train a series of modern Online Hate Detection techniques for a binary classification task. Google's transformer BERT was found to significantly outperform the other techniques (F1 = 0.917 in the test set comprised of both Wikipedia and Twitter data), showing its ability to generalise well across the different datasets. The BERT model was selected for Irish tailoring and was retrained by identifying the most significant hate words contributing to the model's decision to determine hate within the datasets. The identified words were then replaced with Irish hate words at variable rates ranging from 25-100%, and BERT was retrained with the data at each rate. The tailored Irish BERT model showed the greatest performance when 50% of the most potent hate words were replaced with Irish hate words, outperforming the baseline BERT model by 1.7% with an Irish test set, showing that the model was optimised with Irish data. This research provides exploratory analysis into geographically tailored Online Hate Detectors for use in future research and provides some guidance on how to localise Online Hate Detection.

Chapter 1: Project Specification

1.1 Problem Statement

This research paper intends to create and propose an inventive Online Hate detection mechanism using modern approaches to accurately identify hateful discourse in Ireland by using data sourced from online social media platforms. Specifically, this task aims to synthesise a model with the following attributes:

1. *Bespoke for Irish use* – tailor-made and optimised for use in Ireland.
2. *Modern* – underpinned by state-of-the-art natural language processing techniques.
3. *Generalised* – not limited to one social media platform.

The first characteristic explores a research gap wherein a model is optimised to perform better in a specific geographic context, in this case, Ireland. The second characteristic ensures that the model uses state-of-the-art Online Hate Detection techniques to maximise performance. The third attribute intends to ensure the model's transferability across different social media mediums, as most approaches in related literature only involve single platforms.

On balance, this paper attempts to create a foundation for optimising modern Online Hate detection mechanisms in Ireland that can be improved upon and applied to other geographic locations to benefit future work.

1.2 Background

The spur of social media creates an online environment where people can communicate and express themselves freely. However, the growth of online communication results in the proliferation of Online Hate and other forms of abuse, which concomitantly invented the field of Online Hate Detection – a notion predicated on utilising statistical-based, machine learning classification techniques to determine the sentiment of online text to identify and extirpate hateful discourse from online communities. The Pew Research Centre evidence the severity and abundance of Online Hate, showing 40 % of Americans experienced online harassment, and 62 % *"consider it a major problem"* [1]. This field also has theoretical underpinnings in numerous other fields, including social psychology, that often study the consequences of Online Hate, showing that it has numerous detrimental effects on victims' health [2]. Not only does Online Hate have an immense impact on people, but its efficacy also extends further, even affecting policy and decision making procedures at a conglomerate level [3]. For these reasons, the necessity of Online Date Detection protocols is apparent, and the influence hate has on the community is significant and ever-growing. There are also intrinsic cultural and geographical factors that nuance Online Hate Detection; types of hate have been disparate and often inexact in different geographic contexts [4].

The potential for optimising modern hate detection solutions based on geographic and cultural qualities is grounds for helpful research.

1.3 Related Work

Altogether, Online Hate Detection is an enormously researched field. Firstly, researchers have made progress conceptualising the notion of Online Hate as a whole – for example, refining the inherently controversial definition of Online Hate to satisfy multiple popular definitions [5], and also measuring the consequences Online Hate has on the mental health of victims, which is very detrimental [2]. Moreover, the fundamental difficulties that hinder Online Hate Detection models are thoroughly examined by researchers, showing their root causes and methods to overcome them. For example, the problem of *polysemy* hinders natural language processing as a whole and especially in the case of Online Hate Detection [6], but has shown to be somewhat limited with the implementation of different embedding techniques [7]. Similarly, the efficacy of different hate classification techniques are thoroughly studied in multiple works of literature, marking the evolution of more primitive and less effective approaches, like keyword-based [8], to more innovative and state-of-the-art approaches, like artificial neural networks, which consistently outperforms baseline models [9]. Research improving the generality of modern detection techniques is a recently studied facet of Online Hate Detection. Indeed, most research pertains to only one social media domain which hinders the solution's transferability. However, recent studies have shown the potency of multi-platform approaches, resulting in a very high degree of success [5]. However, very little work has been done to utilise cultural and geographic qualities to optimise a model for a given country, creating a very apparent research gap. As such, the research conducted will be based on the related literature in order to develop a solution adequate to satisfy the goals set out in the Problem Statement (see §1.1).

As a whole, the related literature will be used to help define and construct an intuitive definition for Online Hate, identify and resolve common difficulties of Online Hate Detection, utilise the most modern approaches found by researchers and contribute towards research gaps in geographic-based hate detection.

1.4 Resources Required

- Python version 3.8.8 for programmatic implementation.
- Jupyter Notebooks version 6.3.0 for Python implementation and analysis.

Chapter 2: **Introduction**

Many techniques have been implemented to combat Online Hate Detection with incredible success. However, the synthesis of an Online Hate Detector engineered for a specific geographic context is a vastly unexplored area of research. This research paper explores possible solutions for creating an effective detector for use in Ireland. Indeed, this approach aims to explore solutions for detecting hate in Ireland by creating a model that is comprised of three important characteristics:

1. *Bespoke for Irish use* – tailor-made and optimised for use in Ireland.
2. *Modern* – underpinned by state-of-the-art natural language processing techniques.
3. *Generalised* – not limited to one social media platform.

The approach aims to incorporate data from multiple social media platforms to increase the model's ability to generalise; to perform well with different kinds of unseen data. A model that can generalise means that its performance is less susceptible to decrease when faced with different data platforms. This is an essential characteristic for an Online Hate Detector, given that the online realm is comprised of hundreds of different social media platforms. Moreover, the approach intends to compare the efficacy of different modern hate detection strategies like deep learning, word embeddings and distributional semantics, which have been shown to provide state-of-the-art results in related literature to find the method that maximises the performance of the task. Lastly, the approach explores different mechanisms of tailoring the model for Irish use. The primary mechanism explored in this research was to identify significant hate words in the training data by determining the importance of the words in the model's decision-making – the most significant words were chosen to be replaced with Irish hate words at different rates in order to retrain the highest performing model with Irish hate terms in order to optimise it for use in Ireland.

Whilst the research aims to tailor a model for Irish use, the scope pertains solely to the English language, and the possibility of this research extending to Irish Gaelic is outwith the scope of the project. Moreover, the research only considers detecting hate in written mediums – it is assumed that the model will have no mechanism for detecting hate from online videos or audio files.

Ultimately, this research does not intend to propose an infallible Irish Hate Detector but instead explores the different mechanisms for tailoring a model for use in a specific geographic context whilst still being a modern and generalised approach for Online Hate Detection. The models used in this research will be freely available with the intent to open further discussion and exploration into Online Hate Detection in bespoke geographic scenarios.

Chapter 3: Related Work and Ideas

This chapter encompasses the related literature and ideas needed to presuppose this research topic. Firstly, §3.1 researches the concept of Online Hate through a theoretical lens and why it is vital to have an intuitive understanding and denotation to base this hate detection approach. Secondly, §3.2 analyses the common difficulties which languish hate detection as a whole across different research papers to devise the most prudent techniques to limit them. Moreover, the different classification techniques utilised in Online Hate detection are researched in §3.3 to identify the most successful approaches posited by related researchers. Also, the influence of geographic-based factors identified by related work is explored in §3.4 to convey the fruitfulness of a bespoke Irish approach to Online Hate detection. Lastly, the overall research is discussed in §3.5 to show how the knowledge obtained from the related work will act as a foundation for creating a bespoke model optimised for Irish use that adheres to the three prerequisite characteristics set out in the Problem Statement (see §1.1).

3.1 Conceptualising Online Hate

3.1.1 Defining Online Hate

Foremost, the application of Online Hate Detection is faced with objectively deducing what constitutes *Online Hate* (specifically *Online Hate speech)*. This notion is inherently nuanced, with its definition having roots in several academic fields and conceptual frameworks. The difficulty in defining a concrete definition is ubiquitous across multiple literature [10] [11] [12], with social and cultural factors exacerbating subjective ambiguity [13]. There is a struggle to find a common denotation that can satisfy all possible and subjective instances of Online Hate speech. Many studies attempted to define hate speech from their interpretations – some condensed hate speech to deplorable epithets targeted at specific groups and domains, e.g. *"Language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group"* [14] whereas some others diluted the definition; pertaining to all offensive language and intolerable discourse, e.g. *"Offensive name-calling, purposefully embarrassing others, stalking, harassing sexually, physically threatening, and harassing in a sustained manner."* [15] Salminen *et al.* synthesised a definition of *Online Hate* which built upon the core sentiments of collated definitions across multiple common sources:

"Online Hate is composed of the use of language that contains either hate speech targeted toward individuals or groups, profanity, offensive language, or toxicity – in other words, comments that are rude, disrespectful, and can result in negative online or offline consequences for the individual, community, and society at large." [5]

This captured the central values of the collated definitions, which they describe to be *"(a) use of language, (b) targeting, (c) individuals or groups, and (d) hate being action with consequences at various levels of society"*.

As such, this cast a broad and relatively concrete typology of *Online Hate*, which satisfied many familiar interpretations of it and presents an intuitive denotation that will underpin this research

paper.

3.1.2 Consequences of Online Hate

The effects of Online Hate are categorically adverse, with calamitous consequences on the health of online communities. In the field of social psychology, the adverse effects caused by Online Hate and its correlation with one's mental state are seemingly indubitable. Indeed, the effects of traditional bullying, whether physical or emotionally executed, have axiomatic consequences on the victim's mental health. Studies in this field indicate that the negative effects caused by traditional bullying are comparable, if not superseded by cyberbullying – Wang *et al.* compared the levels of depression among traditional bullies/victims and victims of cyberbullying and concluded the latter showed higher rates of depression [16]. Similar studies have also described cyberbullying not only elevating rates of depression but also worsening one's body image over time [2]. This all highlights the pervasive implications Online Hate creates as it plagues online spheres and shows why many implement Online Hate detection frameworks to help combat it.

3.2 Difficulties with Online Hate Detection

Table 3.1: Difficulties with examples

Difficulty	Example
Polysemy	1) Jeez, they've been around the globe, they're definitely travellers 2) All Irish people are travellers
Informal Language	Look at all these ppl
False Positive	I hate going to school every day
False Negative	I hate white people
Obfuscation	I h*te wh1tep30pl3

Being a subset of natural language processing, using machine learning approaches to derive meaning from text is already considered to be a challenging task in the field of artificial intelligence and linguistics [17]. As such, narrowing the lens to specifically processing the malicious intent of text presents a plethora of additional challenges. Work has been done to study the difficulties and the best ways to combat them. The first problem that complicates Online Hate detection is *polysemy*. This term represents the ambiguity of the meaning of words, whereby words can have different meanings in different contexts [6]. This poses severe complications for any hate detection technique *viz.* oftentimes, it is not as simple as categorising an expression as hateful if it contains a potentially hateful term. For example, observing Table 3.1, the first instance of the word "traveller" is used to denote someone touring or sightseeing different places around the world, but in different contexts, it can be used as a derogatory term to discriminate against Irish people, as evidenced in the second example. This problem is inherent in virtually every language, as long as ambiguous terms exist. Nonetheless, different protocols have been studied to combat polysemous words in the text for natural language processing. For example, Naseem *et al.* compared different word embedding techniques for refining ambiguous language and resolving the context of polysemous Tweets [7]. They found that Deep Context-Aware Embedding performed significantly better than existing techniques for handling polysemous words. This shows that despite the existence of polysemy, there are successful methods to combat it.

The second challenge is using unstructured and informal language (mainly present in online text). In Table 3.1 this is highlighted with the word "ppl" used to represent the word "people". This increases the noise data and affects model performances. However, deep learning models fine-tuned on informal data have shown to work well with informal language and can be further increased with data annealing techniques [18].

Subjectivity is another fundamental concern in Online Hate Detection, particularly regarding labelling truly hateful data. As aforementioned, the definition of Online Hate is often convoluted, with many different interpretations; it becomes increasingly difficult to construct adequate training data with ground truth labelling. Different approaches have been used to reliably label training data, for example, Vidgen *et al.* used *"trained annotators over four rounds"* to label their dataset [19] . Using trained annotators and multiple rounds of labelling limits the inherent subjective bias and allows the labelling to be more objective to the agreed definition.

Another issue is the lack of diversity of data in recent studies – analysis of single-platform data (especially from Twitter) is overabundant. This reduces the generality of many models created in different studies because the characteristics of textual data, be it size, style and context, can vary on different social media platforms. For instance, Twitter has a character limit of 280 characters and Reddit a limit of 40k. This disparity infers that a model trained on wholly Twitter data is likely to perform poorly on non-Twitter data due to the contrasting characteristics. As such, the implementation of multi-platform hate detection mechanisms provides better generalisation and scalability, as shown by Salminen *et al.* who created a classifier to identify hate across multiple platforms, including Wikipedia, YouTube, Twitter, and Reddit [5].

The final challenges posited by Online Hate detection are false positives/negatives. False positives are mislabelled samples of text, wherein a non-hateful expression is inaccurately labelled as hateful. Table 3.2 depicts an example of a false-positive – a potential classifier could understandably mislabel this term as hateful because it references a potentially hateful term. This is because training data often overemphasises slurs as hateful regardless of when they are used [14] and other research has shown that identity terms, e.g. *"gay"* often result in false-positive bias [20]. False negatives are truly hateful samples that are inaccurately classified as non-hateful. This is made worse since intuitively obvious true positives have been noted to elude detection from intentional spelling errors, using leetspeak[1] and modifying white-space [21]. This is known as text obfuscation. However, false positives/negatives rates have been shown to improve with more sophisticated embedding/classification techniques [5].

There are numerous challenges with Online Hate detection, including polysemy, informal language, false positives/negatives and obfuscation, but as research in the field progresses, so do the techniques used to combat them.

3.3 Hate Detection Techniques

3.3.1 Simple-lexicon

Numerous different classification techniques have been utilised for hate detection purposes over time. Very primitive frameworks, like simple-lexicon, i.e. keyword-based approaches, were the first to be conceived as a potentially viable solution for Online Hate detection. Lexicon-based approaches involve either the construction of dictionaries that store keywords that represent semantically negative sentiment or the synthesis of corpora that captures contextual and syntactic information of

[1] Leetspeak: using numbers to mimic letters

the keywords, e.g. co-occurrence patterns. An example of this approach in practice is reflected via Gitari *et al.* who created a rules/lexicon-based approach for hate speech detection by constructing their lexicon of expressions using semantic and subjectivity features in order to build a hate speech classifier [22]. For racist/non-racist classification, their results concluded with a 70.83 and 69.85 % F-score on their first and second corpora, respectively. Similarly, Davidson *et al.* used keywords from Hatebase and hateful phrases from internet users to query the Twitter API to synthesise a dataset [14]. The dataset consisted of 25k Tweets from the API – labelled by CrowdFlower workers into three categories: *"hate speech, offensive but not hate speech, or neither offensive nor hate speech"*. Their best performance showed an F1 score of 0.90 but noted that *"almost 40 % of hate speech is misclassified"*. This conveys the limitations of keyword-based approaches in hate classification, particularly for irregular forms of hate. Indeed, the limitations of such simple lexicon-based approaches are ubiquitous. For example, an approach predicated on a dictionary of hateful-lexicon is thereby limited by the content of the dictionary – new terms and derogatory phrases are invented every day, which means it would have to be frequently updated [8]. As such, this form of hate classification, despite having moderate success, is superseded by more inventive approaches.

3.3.2 Distributional Semantics

Another category of hate detection mechanisms is *distributional semantics* based techniques. These are highly sophisticated compared to the aforementioned simple-lexicon frameworks; they rely on word usages, primarily the meaning behind phrases and expressions. Specifically, distributional semantics techniques involve word vectorisation, word embedding and forms of latent semantic analysis. Word embedding is popularly employed to encode the connotations and meaning of words, grouping them close together in the vector space to convey that they share similar meanings. Popular instances of word embedding and vectorization include *term frequency* (TF) and *term frequency – inverted document frequency* (TF-IDF). The former represents the frequency of a word within a document – the latter is an extension of this. However, it aims to find the importance of the word, with the underpinning implication that a word with a higher frequency will tend to have a greater impact or contribution to the sentiment of the expression or sentence. The success of word embedding techniques for hate detection protocols are well known. For example, Salminen *et al.* collected over 5k manually labelled expressions from YouTube and Facebook and postulated that they had success with Linear SVM using TF-IDF features in order to classify the expressions using multiple labels [23]. Other types of word embedding techniques have likewise proven to be successful – *sentiment specific word embedding* (SSWE) was leveraged by Tang *et al.* to encode the sentiment of different words for Twitter sentiment classification [24]. They postulated that incorporating SSWE with hand-crafted feature sets had the best performance relative to other popular classification techniques. This demonstrates that incorporating distributional semantics-based techniques is successful in hate detection and ameliorates primitive lexicon-based approaches.

3.3.3 Deep Learning Approaches

Deep learning marks the proliferation of more inventive renditions of Online Hate classification, mainly through neural networks. Since their emergence, neural networks have been praised for being efficacious in virtually all aspects of natural language processing [25]. When associated with text analysis, these tend to fall into two categories: convolutional neural networks (CNN) and recurrent neural networks (RNN). The success of CNN in Online Hate detection is well documented. For example, Gambäck and Sidkar used different CNNs to classify Tweets into four categories: sexism, racism, and neither – their results showed that all CNNs outperformed the traditional logistic regression classifier with character n-grams used by Waseem and Hovy in 2016 [26]. Similarly,

RNN's success in Online Hate detection is evident – for example, Saksesi *et al.* used an RNN to automate hate speech detection with a very impressive result [27]. In addition to their innovative architecture, neural networks are very flexible as they have many hyperparameters that can be tuned and are not limited by text features. Much research on the performance of neural networks in hate detection involves adjustments to these models to maximise success. In the same study, Gambäck and Sidkar compared the performance of four different CNN models for Online Hate classification, the first using random vectors, the second using word2vec, the third using character n-grams and the fourth using word2vec and character n-grams. They deduced that the CNN set up with word2vec performed the best [26]. Neural networks are not necessarily limited by text features but can also utilise multiple different kinds – for example, user features have been shown to provide high performance. This is corroborated by Pitsilis *et al.* who investigated the performance of an ensemble of RNN with users' bias towards sexism, racism and tendency to post hateful messages as features [28]. They used this model as a mechanism for discerning Online Hate with Twitter data, training it with over 16k Tweets. They posit that this model outperformed other cutting-edge approaches. Hence, artificial neural networks are very potent instruments for Online Hate detection due to their inherent complexity and the multiple ways to configure and tune the model for optimal performance.

Other modern deep-learning architectures are *transformers*. These are modern algorithms that are based on dynamic word embedding and has been incredibly successful in natural language processing [29], especially hate detection [30]. The success of these models is predicated by how they remove sequential dependency from the text – allowing much greater parallelisation than other deep learning architectures. A fine example of transformers is BERT (Bidirectional Encoder Representation from Transformers), which has been shown to have phenomenal results in natural language processing tasks [31]. BERT has been incorporated in studies as a modern way to implement feature representation; for example, Salminen *et al.* used BERT in the feature transformation part of their study and noted that it outperformed other feature transformation techniques, like Bag of Words and TF-IDF. As such, this modern approach for feature representation will be utilised in this research study to ensure the model uses modern deep learning techniques.

Finally, deep learning approaches also offer the potential for explanation with their results. This is known as *explainable artificial intelligence* (XAI)[32]. This aims to explain why a particular model arrived at a classification conclusion. XAI has been shown to synergise well with hate detection problems [33], rendering it a useful tool for providing exploratory analysis on a bespoke Irish model. Indeed, Wang *et al.* used XAI techniques like feature occlusion to identify important keywords that influenced model decision [34].

These deep learning approaches present a generative framework that can be utilised to create a modern Irish hate detection mechanism.

3.4 Geographical-based Hate Detection

Language is incredibly diverse – there are hundreds of different languages pertaining to different geographical locations around the globe. Furthermore, each language and geographic location brings about its own unique culture. Even within the same language, there is abundant diversity, be it colloquialisms, slang terms, or culture. On the other hand, these diversity and cultural differences have irrefragable implications for hate speech detection. Indeed, research is heavily dominated by English-based hate detection and focused on mono-lingual scenarios. Multi-lingual hate detection mechanisms are poorly researched in comparison, often only incorporating one additional language [35][36]. This means prevalent hate detection methods are not generalisable to a large proportion of the globe. Furthermore, this might seem paramount in an Irish context

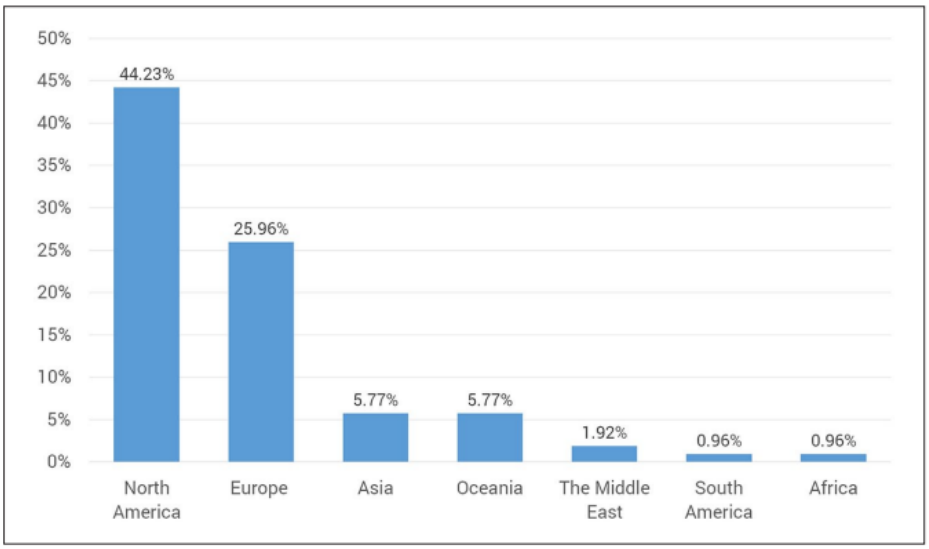

Figure 3.1: Percentage of studies examining different geographic regions, from [4]

due to Irish Gaelic being a secondary language of the nation. However, there is an evident research gap, as no material currently exists related to Gaelic-based hate detection. That withstanding, this research will not focus on resolving this research gap due to the lack of Gaelic-based data used for hate detection training and will instead focus on English-based approaches.

The cultural diversity within the same language is still a central focal point – research frequently notes the importance of geographical factors in terms of Online Hate disparity. For instance, Mondal *et al.* analysed the hate categories and targets of their Whisper dataset for the United States, United Kingdom and Canada – they note that geography can dynamically change the hate categories and hate targets across these nations [10]. Thus, it is evident that geography has an impactful bias toward Online Hate, implying that Ireland will also have its own unique geographical bias. Using existing models not trained within Irish contexts will likely be biased to misrepresented hate categories and targets. This is concomitantly true because models tend to overfit the frequent hate categories/targets present in the training data [20] rendering them less likely to identify less frequent and new forms of hate. This being said, the research toward creating a hate detection mechanism to better capture Ireland's geographical features is virtually non-existent – research primarily encompasses British and American data, thus over saturating the field. Indeed, Matamoros-Fernández and Farkas measured the distribution of geographical breadth of recent studies in hate detection, showing that North American contexts heavily dominate it [4]. Figure 3.1 depicts the distribution, showing that almost half of the studies rely on North American datasets. Moreover, they also elucidate that almost half of the European distribution is focused on the United Kingdom. This further evidences the preclusion of Irish related literature in the field and shows that current hate detection mechanisms are not optimised for Irish use.

3.5 Discussion

The related work in the field of Online Hate detection provides a valuable foundation of knowledge for this research paper.

Work done by researchers into common challenges that are fundamental to Online Hate Detection provides practical techniques on how to limit these difficulties throughout this research paper. Re-

searching and ultimately choosing the formal definition of "Online Hate" is essential and beneficial to the research as it is used to define the scope of what data samples pertain to the "hateful" or "not hateful" categories. The preponderance of studies comparing the success of different Online Hate Detection techniques provides beneficial insights to this research to choose techniques that have shown the most success in the past. However, in terms of research regarding geographic-based hate detection, a clear research gap exists. Despite studies acknowledging the impact of geographic and cultural factors in hate detection, the actual implementation is sparse. This leaves room for new research towards creating a bespoke model better suited for Irish contexts – leaving this research gap as a fundamental research question of this paper. However, this research gap complicates the research drastically. There are no pre-existing empirically tested techniques for tailoring a model to a geographic context, meaning the approach for tailoring the model was not predicated on any prior research.

Ultimately, the research has shown the importance of Online Hate Detection. In addition, the research showed an apparent disagreement across literature about the formal definition of "Online Hate", leading this research to adopt a comprehensive and encompassing definition. Researching the performance of techniques depicted Deep Learning, Word Embeddings and Distributional Semantics techniques to generate the most successful results in Online Hate Detection. This research will use and experiment with these techniques to maximise performance. Lastly, the evident research gap in localised hate detection complicates this research, as there are no prior studies to base the approach from.

Chapter 4: Project Approach

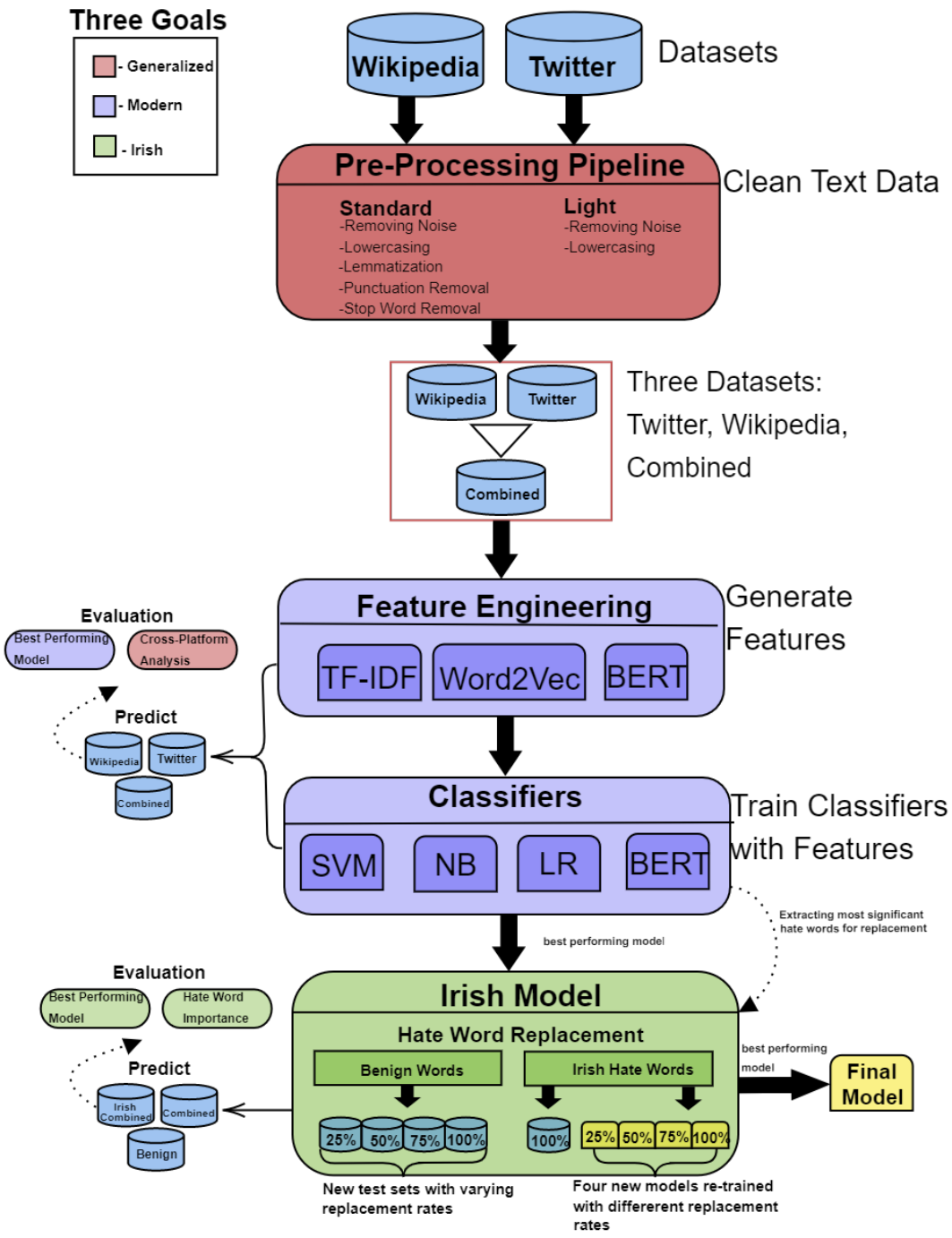

Figure 4.1: Project Approach

This Chapter details the approach and structure to design, implement and evaluate the three main goals of this research. The goals of the research were to create a model that is "Bespoke for Irish Use", "Modern" and "Generalised". Specifically, this Chapter provides a high-level overview of tasks and provides detailed implementation and evaluation methodologies for each of the main goals about the tasks.

High-Level Overview of Tasks

Figure 4.1 depicts the approach taken in this project, highlighting the three goals, the high-level implementation steps, and wherein the approach they were evaluated. The Figure can be broken down into individual tasks:

1. Collecting and cleaning datasets
2. Feature Engineering
3. Classifiers
4. Irish Model

These tasks pertained and/or overlapped with one of the three main goals. The Figure is colour coded to infer what goal they refer to. The goal of Generalisation was encompassed within the tasks of collecting and cleaning the datasets and evaluating model performance, specifically cross-platform analysis. The Modern goal pertains specifically to the Feature Engineering and Classifier tasks and is likewise evaluated on model performance. The goal of optimising the model for bespoke Irish use, was executed within the Irish Model task, which involved retraining the highest performing model from the Modern section with Irish hate vernacular to optimise it for Irish use. This goal is evaluated both in model performance and by how critical it used hate words in decision-making.

4.1 Generalization

This section outlines the approach in implementing and evaluating the goal of "Generalisation" or having a solution that uses cross-platform data to ensure that the model can generalise across different data sources.

4.1.1 Cross-Platform Data

For this research, it was essential to use datasets from multiple different social media platforms. Cross-platform data in Online Hate Detection has shown to increase the potency of model generality – performance on different kinds of unseen data [5]. Conversely, cross-platform data escalates the complexity of the task significantly [37] and so deducing the number of datasets to use was difficult. Given the time frame of this research, it was decided to use two datasets with vastly different characteristics of text data. Due to this, Twitter and Wikipedia stood out as contrasting platforms with contrasting text data. For example, Twitter has a far shorter character limit than Wikipedia, meaning that the Twitter data consisted more of shorter samples of text that were more to the point. In contrast, the Wikipedia data was comprised of longer text samples (for in-depth comparisons of the data and data considerations, see 5.1).

4.1.2 Evaluating Generalization

In order to evaluate the generalisation of the solution, a simple paradigm was synthesised. Firstly, the cross-platform data created three datasets: 1) Twitter, 2) Wikipedia, 3) Combined. Then

each chosen Online Hate Detection technique would be trained with each of the datasets. One model specialised for Twitter, another specialised for Wikipedia, and finally, one trained with a combination of both platforms. Figure 4 depicts where this goal is evaluated – each model would then predict on each of the validation sets. For example, the model trained with Twitter data would be used to predict on the Twitter, Wikipedia and Combined validation sets to generate three results. These results were then compared with each other as a measurement of cross-platform analysis to see how each model performs on its own validation set and the others. A model would be generalised if the model's performance did not vary much between the different sets.

4.2 Modern Techniques

This section elaborates on the approach for incorporating modern Online Hate Detection techniques into this research. The benefit of using modern techniques is to maximise the performance of the approach. This stage is broken down into two parts: 1) Feature Engineering, 2) Classifiers. The Feature Engineering process involves creating features from the text data so that classifiers can interpret it. The Classifiers refer to the machine learning classification techniques that interpret the features to make a classification decision.

4.2.1 Feature Engineering

Feature Engineering is a fundamental and crucial step in any supervised machine learning task. Machine learning models cannot understand text data without it being encoded in some way. There are numerous methods for creating features from text data, and many were outlined in the Related Work section (see §3). For this research, three thriving and modern techniques were leveraged in the Feature Engineering process: 1) TF-IDF, 2) Word2Vec, 3) BERT. This encompasses a more naive distributional semantics approach, a word embedding technique, and a transformer-based deep learning technique. For more information regarding the choice of these feature engineering techniques, see 5.2.

Using modern Feature Engineering Techniques in this task was a crucial step and indubitably enhanced the overall performance of the models.

4.2.2 Classifiers

The classifier is the end model that interprets the features provided in the Feature Engineering stage to make predictions. Classifiers are usually underpinned by mathematical frameworks and theorems that manipulate the features to generate a probability. As such, different classifiers were explored within this research. The choices of classifiers are a critical step, and it was intended that modern and successful classifiers be used to maximise the performance of the approach. In this research, both traditionally successful and more nuanced deep learning classifiers were used and compared. The chosen classifiers were: 1) Logistic Regression, 2) Support Vector Machines (SVM), 3) Naïve Bayes, 4) BERT, due to their abundance in related literature [5] [38]. For more information regarding the choice and description of these classifiers, see 5.3.

Using modern and successful classifiers undoubtedly benefited this research as the efficacy of the classification algorithms is one of the most critical parts of any supervised learning task. It

maximised the potential insights and performance of the approach as a whole.

4.2.3 Evaluating Modern Techniques

The evaluation of this goal is depicted in Figure 4.1, where the modern Online Hate Detection techniques were used to predict on the test datasets and were evaluated using the F1-Score metric. The list of techniques that were evaluated can be seen below:

- TF-IDF Logistic Regression
- TF-IDF Naïve Bayes
- TF-IDF SVM
- Word2Vec Logistic Regression
- Word2Vec Naïve Bayes
- Word2Vec SVM
- BERT

As it can be seen, each classifier was used with a combination of the TF-IDF and Word2Vec features, whereas BERT was modified to be used as a classifier and used its own features. The performance of all the techniques were compared using the evaluation metric, where the highest performing model was utilised for Irish tailoring.

4.3 Bespoke Irish Model

This section discusses the approach for creating a "Bespoke Irish Model" tailored and specialised for use in the Republic of Ireland, as ideated in the Problem Statement (see §1.1). As described in the previous sections, the highest performing model was then selected for this research. Tailoring a model to a specific geographic region has been a sparse and unresearched topic within Online Hate Detection. As such, no pre-existing paradigm was adopted to conduct this research, so different ideas were explored when tailoring the solution for use in Ireland.

4.3.1 Replacing Hate Words

Irish Hate Words

The first idea for tailoring the solution was to identify the most common hate words used within the data and replace them with hate words that are more commonly used in the Republic of Ireland and retrain the model with the new data. This is a plausible hypothesis because the definition of Online Hate used within this research encompasses the use of hate words and profanity. Hence, hateful nouns present within the data, e.g. 'bitch', could be replaced with other nouns to create a dataset with samples of Irish hate artificially.

However, there existed some nuances that complicated this task. For example, if all of the hate words were replaced with the Irish hate words, the patterns and insights from the original data would be lost. Indeed, if all instances of 'bitch' were replaced with an Irish hate word, the model would no longer see 'bitch' as a hate word. Hence, it was a cautious task to incorporate the new Irish hate words within the model so that the model could interpret Irish hate without losing the capability of identifying the general and more common types of hate that were present within the original dataset. Due to this, different replacement rates were experimented with – ranging from 100 %, 75 %, 50 %, 25 % replacement rates. For example, in the 100 % replacement model, all of the instances of the hate words were replaced with Irish hate words. This created four new models trained with the different replacement-rates that could then be compared. Figure 4.1 depicts these four new models within the Irish Model goal.

Benign Words

Replacing the exact identified hate words with benign words was another idea for examining the efficacy of the Irish model and seeing how important the model deems the hate words in determining a class label. Instead, this would not be used to retrain the models, but the replacement would occur only within the test sets to be predicted. It was reasoned that if the model does not drop performance significantly when the hate words in the test set are replaced with benign words, then the model still maintains the capacity to understand the hate from context and not only from hate words. Similar replacement rates were used to replace the hate words within the test set with benign words. This created four different validation sets, one with 100 % replacement, another with 75 etc... The performance of the Irish model could then be used and compared across the four validation sets.

4.3.2 Evaluating the Bespoke Irish Model

Evaluating Hate Word Replacement with Irish Hate Words

Evaluating the efficacy of hate word replacement with Irish words was relatively straightforward. It ultimately evaluated which model retrained with the different replacement rates performed the best between the original test set and the test set replaced with Irish hate words. Indeed, it was aimed that the Irish model would retain the performance captured from the original data and perform well with a dataset composed of Irish hate. The four different models were used to predict against the original test set, and the test set was replaced with Irish words. The model that performs the best across both datasets would be deemed to have the most precise balance of replaced hate words.

Evaluating Hate Word Replacement with Benign Hate Words

This idea was mainly employed for evaluating the Irish model's ability to determine hate without the presence of the most common hate words within the dataset. The Irish model was used to predict the four new test sets with varying word replacements to see the effect removing hate words from the test set vocabulary has on the performance. It hypothesised that the model would face a significant loss in performance because the presence of hate words is a valid facet for deducing hate. However, suppose the model could also determine hate from other characteristics, like semantic relationships and context. The performance should not drop as much as a more naive classifier that is more heavily dependent on keywords, like TF-IDF. As such, the model was compared against the TF-IDF models in order to compare this ability.

Chapter 5: **Implementation**

5.1 Datasets

This section discusses the approach taken into collating and pre-processing the datasets used within this research.

Table 5.1: Dataset Characteristics

Dataset	No.Samples	Avg Words	Avg Characters	Avg Uppercase	Avg Punctuation
Twitter	24783	15	85	5	1
Wiki	159571	69	394	17	8
Wiki Balanced	31816	63	352	29	7
Combined	184354	61	352	16	7
Combined Balanced	56599	42	235	18	5

Table 5.2: Data Class Distribution

Dataset	No.Samples	No. Hateful	No. Not Hateful
Twitter	24783	20620 (84%)	4163 (16%)
Wiki	159571	15908 (10%)	143663 (90%)
Wiki Balanced	31816	15908 (50%)	15908 (50%)
Combined	184354	36528 (20%)	147826 (80%)
Combined Balanced	56599	36528 (65%)	20071 (35%)

5.1.1 Data Considerations

Adhering to the Project Approach, multi-platform datasets were chosen for this research. Scrupulous choosing criteria were enforced in order to ensure the datasets were apt. The first criterion required the datasets to pertain solely to the English language. It is outwith the project's scope to use multi-lingual data because Ireland's primary language is English. At the time of this research, there existed no formidable datasets for Online Hate Detection purposes in Irish Gaelic.

It was also required that the datasets had robust labelling strategies to mitigate the subjective factors of Online Hate. Each dataset sample was labelled by multiple trained persons employing voting systems where the majority label would be assigned to the sample. The quality of data is paramount in any classification task, and because Online Hate is so subjective, this democratic labelling strategy limits the inherent subjective challenges.

5.1.2 Twitter Data

Description

One dataset that was chosen for this research was the Twitter dataset synthesised by Davidson *et al.* in their 2017 study *Automated Hate Speech Detection and The Problem of Offensive Language* [14]. They leveraged a dictionary of hateful lexicon from Hatebase.org and queried the Twitter API for Tweets containing the lexicon – resulting in 85.4 million Tweets from 33,458 Twitter users. From this large corpus, 25k Tweets were randomly sampled and were labelled by Crowd Flower[1] workers into one of three hate speech categories: "hate speech", "offensive but not hate speech" and "neither offensive nor hate speech". The Tweets that could not be labelled were removed, resulting in a final dataset of 24,802 Tweets. This dataset is made publicly available on their Github[2] for further use in the field of Online Hate Detection. The CrowdFlower workers came to an astounding 92 % agreement when labelling the dataset, and the disagreements were resolved from a majority voting decision.

Benefits

- Provides an overwhelming proportion (84%) of hate samples (see Table 5.2). This is beneficial given that data sparsity is a common problem in classification where not enough of the positive class is given during training. However, the abundance of hate samples means this problem will not occur.
- Used within in previous studies where an excellent F1 score of 90% was achieved [14], showing that the data is apt for generating high-performance results.
- Labelled by CrowdFlower workers to mitigate subjective variance within the labels.

Challenges

- Twitter has a short character limit relative to other platforms (280 characters at the time of research), meaning that the samples of the datasets are relatively short compared to other datasets. Table 5.1 depicts the characteristics of the datasets, where it can be seen that Twitter has a relatively short average number of words and characters per sample (15 words and 85 characters on average). This indicates that users on Twitter are less likely to engage in complex and lengthy discussions. The Twitter dataset on balance, provides short and to-the-point samples of hateful discourse rather than more complex instances of hate in long paragraphs of text in discussions or arguments.

5.1.3 Wikipedia Data

Description

Another dataset that was chosen was the popular Wikipedia dataset derived for the "Toxic Comment Classification Challenge" as part of the Kaggle data science competition[3]. This was a challenge issued by the Conversation AI team, founded by Jigsaw and Google, to use their dataset to classify Wikipedia comments into multiple labels: "threats", "obscenity", "insults", and "identity-based hate". Similarly, the labels for this dataset were deduced by trained CrowdFlower workers

[1] Crowdsourcing service that complete manual AI tasks.

[2] https://github.com/t-davidson/hate-speech-and-offensive-language Last accessed: 06/05/2022

[3] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data Last accessed: 06/05/2022

– 10 workers would rate each sample, and the label would be determined from majority voting. If one signs up for the challenge, the challenge issues both a train and test dataset, publicly available to download.

The definition of "Online Hate" adopted within this paper encompasses all of the labels used within the Wikipedia dataset "severe toxic", "toxic", "threat", "insult" and "identity hate". If a sample was labelled with at least one of them, they were labelled 'hateful'; otherwise, it was labelled 'not hateful'. Moreover, the train and test datasets were cumbersome, 150k samples each, so the training set was only used for this paper.

Benefits

- There's an abundance of data with 159571 samples (see Table 5.1).
- Labelled by CrowdFlower workers to mitigate subjective variance within the labels.

Challenges

- A very apparent class imbalance towards the negative class of 90% (see Table 5.2). This could make it hard for a potential model to derive insights when the negative class overwhelms the positive class.
- It is evident that the Wikipedia dataset is comprised of much longer text samples of an average of 69 words and 394 characters (see Table 5.1), meaning that the text data is much more complex and structured.

5.1.4 Combined Data

The aforementioned Twitter and Wikipedia datasets were concatenated (combined) to make one large dataset intended to enforce generality and transferability. Table 5.1 shows the characteristics of the combined dataset that have characteristics very similar to the Wikipedia dataset in terms of average words and characters, uppercase and punctuation. This is expected given that the Wikipedia data vastly outnumbers the Twitter data.

Benefits

- There is an abundance of data with 184354 samples (see Table 5.1).
- Contains both structures of data: to-the-point and simplistic (from Twitter) and lengthy, complex, and structured (from Wikipedia). This creates a complete dataset where different types of text data are represented so that the model can be trained with both text data structures.

Challenges

- A very apparent class imbalance towards the negative class of 90% (see Table 5.2). This could make it hard for a potential model to derive insights when the negative class overwhelms it.
- Labelled by CrowdFlower workers to mitigate subjective variance within the labels.
- It is evident that the Wikipedia dataset is comprised of much longer text samples of an average of 69 words and 394 characters (see Table 5.1), meaning that the text data is much more complex and structured.

5.1.5 Balancing Data

It is important to note that the Wikipedia dataset is far more significant than the Twitter dataset, and there exists a class imbalance problem within the Wikipedia dataset. The fact there are more Wikipedia samples than Twitter isn't detrimental to the task – data sparsity (lack of quality and quantity of data) is often a major problem of NLP tasks – having more data belonging to either platform will be beneficial for the solution.

However, the problem of class imbalance can be potentially detrimental – this problem occurs when most of the data pertain to the negative class (in this context, 'not hateful'). Class imbalance can result in the model not having enough of the positive class relative to the negative class to capture the patterns to deduce whether a random sample is positive or negative. There exist two methods for mitigating the effects of class imbalance: 'up sampling' and 'down sampling'. Upsampling occurs when samples of the positive class are duplicated so that the dataset becomes roughly equal. In contrast, downsampling occurs when the samples within the negative class are pruned so that the dataset becomes roughly equal. Upsampling in this scenario is imprudent because the class imbalance is so drastic. Observing Table 5.2, it can be seen that the positive class makes up a mere 10 % of the entire Wikipedia set. As such, upsampling the positive class runs a significant risk of overfitting the model because the positive samples will have to be duplicated many times. It was more prudent to create another dataset for the models that could not capture the patterns from the positive class by utilising downsampling on the Wikipedia dataset. Random negative samples were removed from the Wikipedia set such that the number of positive/negative samples was equal. Thus, a new Combined dataset was synthesised using the original Twitter data and the newly defined balanced Wikipedia data. This resulted in a new class imbalance in favour of the majority class. However, this does not create the same problems as before; more of the positive class creates variance and diversity, allowing the more sensitive models to learn the patterns of the positive class.

5.1.6 Relabelling The Data for Binary Classification

The definition of "Online Hate" adopted by this research paper is broad and pertains to anything remotely offensive, including any instance of profanity (see §3.1). This research paper focused on the binary classification problem of categorising the data into two classes: 'Hateful' and 'Not Hateful'. The original Twitter and Wikipedia sets are labelled with multiple labels. As such, the Twitter dataset that was originally labelled with: "hate speech" and "offensive but not hate speech" were relabelled as "hateful", whereas the "neither hate speech or offensive" was relabelled as "not hateful". Similarly, the "severe toxic", "toxic", "threat", "insult", and "identity hate" are all synonymous with this definition of "hateful". In contrast, any samples that were not labelled any of the above were considered "not hateful".

Observing Figure 5.1 below, random samples of the new binary classes from the Twitter and Wikipedia datasets are depicted: The red samples indicate "hateful" samples, whereas the green

Twitter: it's a brand new day and the birds they chirpin

Wiki: You, sir, are my hero. Any chance you remember what page that's on?

Twitter: That shade of faggot clashes with your eye color.

Wiki: eat a dick \n\n and fuck off

Figure 5.1: Random Hateful and Not Hateful Samples

text indicates "not hateful" samples. Unsurprisingly, the hateful samples are both derogatory and

involve hostile intent.

5.1.7 Data Pre-Processing

Text pre-processing is essential in virtually every Natural Language Processing (NLP) task. However, it becomes imperative in the field of Online Hate Detection, as the quality of the text data has a direct relationship with the quality of the prediction of sentiment [39]. A pre-processing pipeline was created to pre-process the data into a suitable format to be interpreted by the models efficiently. Some classification techniques tend to work better with different data pre-processing steps than others, so a dynamic pre-processing pipeline was created where each pre-processing step could be selected or omitted. For the classifiers used in this research (see Classification section), two permutations of the pipeline were used: "Standard Pre-Processing Pipeline" and the "Lightly Pre-Processing Pipeline". The former includes all the pre-processed techniques available within the pipeline. In contrast, the latter precludes many pre-processing techniques to keep the data more similar to the original text.

Noise Removal

Text data noise is defined as *"any kind of difference in the surface form of an electronic text from the intended, correct, or original text"* [40]. Indeed, the original Twitter and Wikipedia datasets were prolific with noise that complicated the classification task. Noise within text data disguises the true and important patterns that the classification models interpret, so this crucial step of removing noise inherently improves the degree of model performance in the classification stages.

The Twitter and Wikipedia data both had different examples of noise. In the Wikipedia dataset, it was common for samples to include the IP address of the user who commented – a clear example of noise because it has no utility in determining a class outcome. Moreover, in the Twitter dataset, retweets were common to be pre-fixed with a long string of exclamation marks. Present in both datasets were references to usernames indicating who had written the comment/Tweet – another example of noise. Other examples of noise present in the datasets were multiple consecutive spaces and new lines. Examples of noise within Twitter and Wikipedia data are depicted below:

"!!!!!!!!! RT @C-G-Anderson: @viva-based she look like a tranny" - Twitter

"I'm sorry I screwed around with someones talk page. It was very bad to do. I know how having the templates on their talk page helps you assert your dominance over them. I know I should bow down to the almighty administrators. But then again, I'm going to go play outside....with your mom. 76.122.79.82" - Wiki

From the above Tweet, almost half of the sentence is noise, including the long string of exclamation marks, the "RT" tag and the username of the user Tweeting. Likewise, the coherent Wikipedia sample ended with the user's IP address. As such, regex was used to combat the aforementioned examples of noise to remove them from the datasets. The complete list of noise removal steps can be seen below:

- Replacing multiple spaces with only one
- Removing new line characters ('\n')
- Removing the string of exclamation marks and "RT" tag before a retweet
- Removing the username of the user Tweeting/commenting

- Replacing URLs with [URL_PH][4] to reduce variance
- Replacing all mentions of other users with [MENTION_PH]

Removing the noise was a crucial step in data pre-processing, and it ultimately rendered the text in a state that better represented the true intention of the original text. The benefit of this was that the models used within the classifications section would be given the highest quality data to interpret.

Lowercasing

The first step was to convert the text to lowercase, as this has been shown to increase model performances in many Sentiment Analysis studies by reducing the dissimilarity of the text [], e.g. the words "House" and "house" ultimately have the same meaning.

Stop Word Removal

Another pre-processing technique used was the removal of' stop words' – this is vernacular that adds little to no meaning to the text, e.g. the words "the", and "a". The efficacy of removing stop words lies in removing words that are not likely to, by themselves, contribute to the prediction of a class label. Sometimes stop words help to create a context that can be interpreted by more sophisticated models (like transformers), but the majority of the naive Online Hate Detection techniques are not comprehensive enough to understand the context of the words. As such, a list of stop words were extracted using NLTK[5] and all instances of the stop words were removed from the datasets.

Lemmatization

Lemmatisation is a word normalisation technique that reduces a word to its 'lemma' or root form. Word normalisation is often implemented through' word stemming' or' word lemmatisation'. Word stemming involves normalising a word to its root word – for example, the word "eating" will generate "eat". Word lemmatisation is a similar normalisation technique but is a bit more nuanced in that it takes the morphological analysis of the word to render the word as its lemma. For example, "was" is transformed to "be". The choice of word stemming vs lemmatisation is marginal and often an unimportant decision to make in Online Hate Detection as both have been used in popular studies with excellent success [10][14]. Ultimately, lemmatisation will always ensure that the resulting word exists in the English dictionary, whereas stemming can produce words that do not exist but have been stemmed nonetheless. Lemmatization was added to the pipeline because it enforces that the resulting lemma will exist within the English dictionary. Many libraries offer lemmatisation, including SpaCy, Gensim and WordNet, but the choice was unimportant and so WordNet's lemmatiser was added to the pipeline due to its ease of use.

Removing Punctuation

Removing punctuation (including emojis) is a pre-processing technique that has shown some improvement in model performances in Sentiment Analysis tasks [41]. As such, this was also added to the pipeline. Similar to stop words and word lemmatisation, inventive models that can interpret context can avail from the inclusion of punctuation. However, because naive models cannot do this well, it was added as a step in the Standard Pipeline.

[4] PH = Place Holder

[5] https://www.nltk.org/ Last accessed: 06/05/2022

Observing Table 5.3 the different pre-processing techniques are depicted where they exist within a given pipeline. The Standard Pipeline includes all techniques, whereas the Light Pipeline only includes noise removal and lowercasing. The Standard Pipeline was designed for models that have historically shown an inability to determine contextual relationships. The Light Pipeline was synthesised to preclude the pre-processing steps that removed contextual information and were intended for the more inventive models (like transformers).

Table 5.3: Pipeline Pre-Processing Techniques

Pre-Processing Technique	Standard Pipeline	Light Pipeline
Removing Noise	x	x
Lowercasing	x	x
Stop Words Removal	x	
Lemmatization	x	
Removing Punctuation	x	

5.1.8 Train and Test Splitting

In order to test the techniques used in this research, the data had to be split into train and test sets. The premise behind this is to use the training set to train the models so that they can interpret and deduce patterns within the training data to make accurate and precise predictions. Splitting some of the data to form a test/validation/'hold out' set is a mechanism for testing how well the model performs on unseen data – if the model has successfully learned well, it should perform well on the test set. However, if it performs poorly, it indicates that the model has been overfitted to the training data.

As such, the train/test split of a ratio of 70:30 was implemented – this is a good rule of thumb and because there is much data, it is reasonable to hold out more data to the test set because it will be more of an indication of model generalisation. Other popular ratios include 80:20 and 75:25. In addition, it was important that the train/test splits maintained the same ratio of classes to keep training consistent. For example, if there were a class imbalance of 70:30, it would be imprudent to do random splitting in case the majority of the imbalanced class ends up in the test set.

SKLearn's train/test split algorithm was used to derive the train and test sets, and the ratio was selected to be 70:30. This algorithm ensures the samples are stratified between the train/test set, meaning that the ratio of 'hateful' to 'not hateful' samples is maintained between the train/test set. Table 5.4 below depicts the train/test splits for all of the datasets:

Table 5.4: Train/Test Set Characteristics

Dataset	No. of Samples	Train	Test
Twitter	No. Hateful	14427	6193
	No. Not Hateful	2921	1242
Wikipedia	No. Hateful	11136	4772
	No. Not Hateful	100563	43100
Wikipedia Balanced	No. Hateful	11040	4868
	No. Not Hateful	11231	4677
Combined	No. Hateful	25481	11047
	No. Not Hateful	103566	44260
Combined Balanced	No. Hateful	25594	10934
	No. Not Hateful	14025	6046

5.2 Feature Engineering

This section discusses the different feature engineering techniques used in this research. This relates to many of the more comprehensive techniques provided in section 3.3. Feature Engineering is one of the most critical facets of classification within any NLP task [42] as it transforms the text data into a set of features that the models can use to make predictions. The quality of the features is directly linked to model performance. Three Feature Engineering techniques were chosen, relating to the increasing level of success found within the related work, including TF-IDF (Distributional Semantics), Word2Vec (Word Embeddings) and BERT (Transformers).

5.2.1 TF-IDF

Term Frequency – Inverse Document Frequency (TF-IDF) is a feature engineering technique and is part of distributional semantics. TF-IDF is simply the calculation of both the frequency of a term, and the inverse of the document freqeuncy of a given text. It is a statistic that conveys how relevant a word in a corpus is to a given text. The term frequency measures the number of instances of a given the word (t), in a document (d). The document frequency (df) measures the frequency of the term (t) relative to the entire set of documents (N). Whereas, the inverse document frequency is simply the inverse of the document frequency – the log of this is often derived to reduce the weight of idf relative to tf. TF-IDF is simply the product of TF and IDF. Mathematically, this is represented below:

$$TF - IDF(t,d) = \frac{count\ of\ t\ in\ d}{total\ words\ in\ d} . \log \frac{N}{count\ of\ t\ in\ N} \tag{5.1}$$

TF-IDF assigns weights to each word based from the above formula in order to determine the meaning of the text and the relevance of the particular word. For example, if a specific word tends only to belong to a specific document labelled 'hateful', the term itself is likely hateful. Ultimately, in this research, TF-IDF is used to identify the most important words of each class ('hateful and 'not hateful').

TF-IDFs potency in determining term significance explains its inclusion in this research. Given that the definition of Online Hate includes profanity implies that TF-IDF would be successful because profanity would only belong to one class, meaning that TF-IDF should be able to identify a lot of the hateful samples based on profanity alone. However, TF-IDF's approach can easily lead to overfitting if the data is not cleaned to a high degree. Indeed, if the data is noisy, then TF-IDF will begin to measure the importance of the noise to a given class rather than the meaningful terms themselves. As such, this research leveraged SKlearn's 'TfidfVectorizer', and it was fit with the

heavily pre-processed data (see §5.1.7) so that as much noise was removed as possible, including punctuation removal and word lemmatisation. The pre-processing steps ensure that the data includes words as their lemma so that words used in different tenses or forms are grouped together because they all have the same meaning. Moreover, TF-IDF does not handle punctuation well, because punctuation tends to add contextual meaning and tone to sentences, whereas TF-IDF would see punctuation as independent words. Likewise, all words were lowercased so that, for example, 'They' and 'they' would be deemed the same term. Lastly, stop-words were removed because they add no meaning to class prediction but have significant term frequencies. The TF-IDF model was fit with the training data and was used to transform both the training and test corpora using the TF-IDF weights found within the training set.

5.2.2 Word2Vec

Word2Vec is another popular technique that can be leveraged for feature engineering. Word2Vec is a Word Embedding technique that transforms words into word vectors. The word vectors attempt to represent various characteristics, like semantic relationships, denotations and context. Numerical representations of words allow for linear algebraic functions to be applied to them, including the cosine similarity, which can be used to determine word similarity. The Word2Vec architecture in particular, pertains to either the Continuous Bag of Words (CBOW) or the Continuous Skip-Gram Model. The former is like a FFNN that attempts to predict an unknown word from surrounding words. The latter involves a neural network where a hidden layer is used to predict the probability of surrounding words from a given word.

In this research, a Word2Vec model was constructed from scratch using the Gensim library. The Gensim Word2Vec model comes with the ability to choose whether it will use CBOW or Skip Gram – the CBOW model was chosen, partly because it is the default parameter but also because both techniques have been shown to perform well in sentiment analysis tasks [40]. The data was first pre-processed with the standard pre-processing pipeline, lowercasing all words, removing stop words, lemmatisation, and removing punctuation. Lemmatization makes the feature matrix more sparse and removes the same words in a different form that ultimately have the same meaning. It was deliberated whether or not to remove punctuation because it can add contextual properties to a sentence. However, in some studies, it has been noted that removing punctuation has yielded success in Word2Vec models [43], so it was decided to remove them as a pre-processing step. The Word2Vec model was then trained with the data, and the embeddings were extracted as features. Extracting the word embeddings involved using the model to generate word vectors for each word in the training set and taking the average of the vector as the final embedding. This generated a matrix of word embeddings that naive models can use.

5.2.3 BERT

BERT is a transformer constructed by Google and has generated state-of-the-art results in Online Hate Detection [5] [31] [38]. BERT's architecture can have multiple encoding layers and self-attention mechanisms. BERT uses attention mechanisms that determine contextual properties between words in a text. BERT's bi-directional functionality removes sequential dependencies from text, allowing the model to interpret the meaning of text from surrounding words rather than traditionally interpreting it from left-to-right, or right-to-left. Observing Figure 5.2 below, BERT's bi-directional architecture is evident. This is relative to other competitive transformers, including OpenAI GPT's unidirectional and ELMo's shallow bi-directional architecture:

In order to use BERT for classification, it is important to understand how BERT represents its features, or in this case, word embeddings that will ultimately be used for classification. BERT

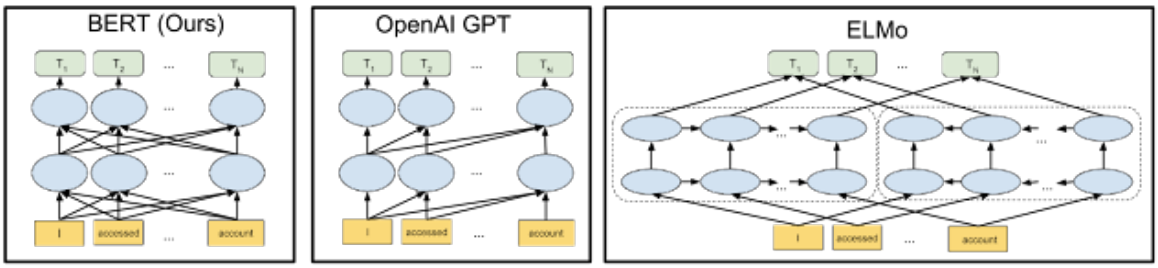

Figure 5.2: BERT Architecture vs. Other Transformers [44]

creates embeddings by combining segment embeddings, token embeddings and position embeddings. Observing Figure 5.3 below, BERT's logic for interpreting input is depicted: The input is a

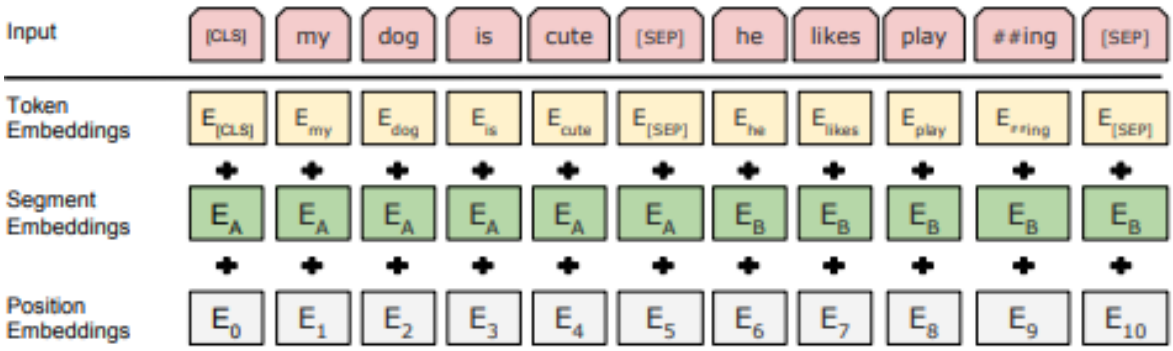

Figure 5.3: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings [44]

sentence – pre-fixed with the "CLS" token indicating the beginning of a sentence and the "[SEP]" token indicating the end of the sentence. The "[MASK]" token illustrates BERT's Masked Language Modelling (MLM), used to understand the context. This process works by masking (i.e. replacing a word with the ["MASK"] token) a random word within the sentence, and then BERT predicts what the masked word would be. The token embeddings are created from a bespoke tokenisation algorithm used for BERT pre-training. The different pre-trained BERT models use different tokenisers. For example, the WordPiece Tokenizer generates subwords and converts them into a 768-dimensional vector in the Token Embeddings layer of the model. The Segment Embeddings represent BERT's need to record the link between a token and a particular sentence – this involves creating two vectors in the Segment Embeddings layer. The Positional Embeddings represent the index of a word within a particular sentence and are encoded into a vector. Hence, the embeddings are the sum of these three embedding strategies and illustrate BERT's feature representation.

In this paper, BERT's inventive feature representation is leveraged for classification.

5.3 Classifiers

This section discusses the classifiers used, a brief description of how they work and how they were tuned for use in this research. Four classification techniques were chosen, ranging from traditionally successful models to more modern Deep Learning models. The traditional classifiers include Logistic Regression, Naïve Bayes and SVM, which have been utilised in many Online

Hate Detection studies [5] [45] and the Deep Learning classifier was an Artificial Neural Network connected to a pre-trained BERT transformer.

5.3.1 Logistic Regression

Uses in Classification

Logistic Regression (LR) is a supervised machine learning classification algorithm that has been incorporated in many Online Hate Detection related studies, with success in binary classification. LR computes a probability of a sample pertaining to a class using the sigmoid function. The function returns a value between 0-1, where a probability closer to 1 represents the likelihood of the sample pertaining to the positive class. LR was chosen as a candidate classifier for this task for many reasons. Firstly, LR is a simple mechanism that takes relatively little training time compared to other classifiers. Moreover, LR assigns weights to the different input features, indicating that some features may be more influential than others in calculating the probability, meaning that the model is very easily interpreted.

Hyper Parameter Tuning

Due to the time constraints of the research and the dataset being incredibly large, computationally expensive methods like Grid Search CV were excluded from the hyper parameter tuning methodology. As such, limited hyper parameter tuning was performed on the classifiers. All parameters were left as SKLearn's Logistic Regression default, aside from one of the most popular parameters to configure, the "C" parameter. The "C" parameter refers to the model's regularisation ability; how it assigns weights to the features. An incredibly large value of C yields better results when there is a high trust within the training data, whereas a low C would be a precaution against overfitting. The default value of 1.0 was changed to 3.0, because scrupulous pre-processing steps were taken on the text data, removing the noise that would render the model prone to overfitting.

5.3.2 Naïve Bayes

Uses in Classification

Naïve Bayes, predicated on Bayes Theorem, is another machine learning classification algorithm. Naïve Bayes is a consistently used classification algorithm across many Online Hate Detection literature, with great success. Bayes Rule is used to go from P(X|Y), where X is the features found from the training set, and Y is the class ('hateful' or 'not hateful) to P(Y|X). Naïve Bayes is an extension to Bayes Rule, for multiple values of X, representing the large number of training features used in classification problems – it has the 'naïve' assumption that all features of X are independent of one another. Furthermore, using Bayes theorem with the multiple X features allows for the probability of Y, based on the features of X to be determined. There are multiple forms of the Naïve Bayes classifier, including Gaussian, Multinomial and Bernoulli. It was prudent to use the Bernoulli Naïve Bayes for this classification task because Gaussian works better with continuous variables, and Multinomial is used for multi-label classification. Ultimately this classifier was chosen because it is a tried and tested traditional classifier and because its widespread use throughout Online Hate Detection and NLP literature.

Hyper Parameter Tuning

Bernoulli Naïve Bayes does not have many parameters to tune, and due to time constraints, all parameters were set to the SKLearn's Bernoulli Naïve Bayes default. Naïve Bayes has achieved high accuracy scores in similar studies without scrupulous parameter tuning [45].

5.3.3 Support Vector Machines (SVM)

Uses in Classification

SVM is another classic classifier within Online Hate Detection. SVM works in binary classification tasks by constructing a hyperplane and assigning each training sample set a point on the plane. The points are arranged in such a way so that a line can be drawn to separate samples belonging to one class from another. SVM aims to maximise the distance between the points in each class (the margin). Unseen data is then assigned coordinates based on its features, and the SVM classifier can deduce the class based on what side of the line it lies. SVM has been shown to work well with text data, especially the linear kernel. As such, SVM was also used as a candidate classifier.

Hyper Parameter Tuning

Similar to Naive Bayes, SVM does not have many parameters to tune. The most critical parameter was to choose the kernel – SVM can have a variety of kernels. However, it is prudent to use the linear kernel for text classification, as many of the other kernels involve transforming the data to higher dimensional planes, which can be highly computationally expensive with large feature sets. Linear kernels also perform the best with high feature sets, which is true in this case when trained with a large corpus of text. Moreover, SVM takes the longest to train out of all of the other traditional classifiers, rendering the option of strict hyperparameter tuning infeasible.

5.3.4 BERT

Uses in Classification

BERT can be used in two ways for classification tasks: 1) extract the pre-trained features and the activations from the layers within the model and use them as features within other classifiers. 2) Fine-tune the last trainable layers within BERT for a specific task, add a dense layer and an output layer, so that the model itself can be used to make predictions. The first method was explored by [5] – after fine-tuning BERT, they extracted the Word Embeddings to use within different classifiers. This paper will explore the second option that fine-tunes the BERT model for classification within Online Hate Detection.

Choosing a BERT Model

Numerous BERT models have been pre-trained by Google and are available publicly to download. BERT intends to be used in transfer learning, where the pre-trained model ought to be fine-tuned to be leveraged in downstream tasks – made possible by its self-attention mechanisms. Out of all the publicly available models, BERT base was deemed the most suitable for this research as it was pre-trained with Wikipedia and BookCorpus data, meaning that it was likely trained with similar Wikipedia data used within this research. The model has 12 layers, 768 hidden layers and 110 million parameters. The model was adopted using HuggingFace – a TensorFlow wrapper used for

transformers[6].

Another BERT model that could have been used instead was BERT large, it is identical to BERT base, but instead, it used 24 encoding layers. However, the BERT large model increases fine-tuning time significantly, and because BERT already takes significant time to train, it was not feasible to use the larger model.

Fine-Tuning BERT

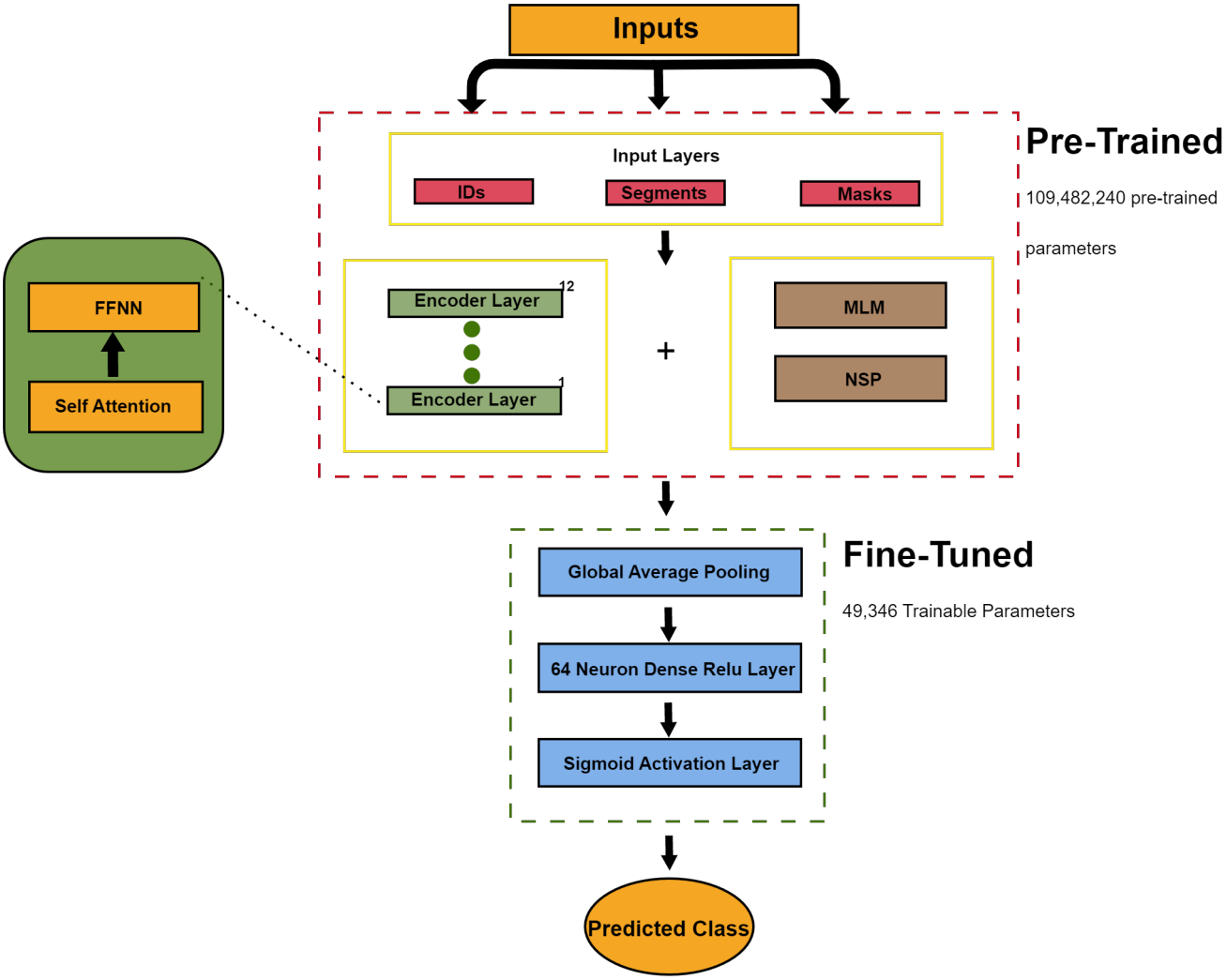

Figure 5.4: BERT Fine-Tuned Architecture

The fine-tuning process first involved lightly pre-processing the balanced Combined dataset. BERT's success is contingent upon its contextual analysis; rendering the original pre-processed data used in the naive classifiers would be unsuitable for this task. Indeed, lemmatisation, removal of stop words and punctuation etc., strips the data of its contextual value that BERT requires to work optimally. Instead, a new dataset was made with a different pre-processing pipeline – the integrity of the original sentence was kept. However, unwanted noise, like IP addresses, usernames etc., that were discussed in the original pipeline were removed. Moreover, BERT requires the text data of the form present in Figure 5.3 – to include special start and end tokens, segments, masks and IDs. A pipeline was created to convert each sample of the data to resemble this form and subsequently was tokenised with the same tokeniser used in the pre-training of the model. This was the

[6] https://huggingface.co/bert-base-uncased Last accessed: 06/05/2022

WordPiece tokeniser and was leveraged by the HuggingFace library.

The chosen BERT model's last 49,346 trainable parameters were then trained with this dataset and added a Global Average Pooling (GAP) layer, a fully connected 64 neuron activation layer and a 1 neuron output layer. GAP has been popularly used to prevent overfitting in CNNs [46]; it works by connecting to the last fine-tuned BERT layer and generates two feature maps (one for each class) and takes the average of the feature maps and passes the resulting vector through to the 64 neuron relu layer. Since there is no parameter to optimise in this layer, the model avoids overfitting at this layer. Furthermore, the 1 neuron activation layer is what generates the final probability for the class decision.

Figure 5.4 depicts both the pre-trained and fine-tuned BERT architecture used within this research. The Fine-Tuned section of the Figure depicts the classification head connected to the pre-trained BERT model, which is ultimately used to generate class predictions.

Choosing an Optimizer and Loss Function

Optimisers are algorithms leveraged by deep learning models to help minimise the loss function or the degree of error within the model. There exists no theoretical guidance when choosing an optimiser [47]. For this study, the ADAM optimiser was used for the fine-tuned BERT model. ADAM has shown to have a significantly lower training cost than other optimisation functions like RMSProp, AdaGrad and SGDNesterov [48], rendering it a viable choice for this research. ADAM works by synergising two different gradient descent techniques: momentum and Root Mean Square Propagation (RMSP). Both techniques are used for optimising gradient descent, and ADAM inherits their potency to outperform other popular optimisation functions.

As aforementioned, the premise of an optimiser is to minimise the loss function. There are numerous loss functions available in deep learning architectures. For binary classification problems, the binary cross-entropy loss function is one of the most popular and for a good reason. Binary cross entropy determines the distance between a predicted probability and the actual class, or more precisely, it is the negative average of the log of the model's predicted probabilities. Due to its widespread use and the fact it is the default loss function, it was used in this study. Other loss functions could have been used, like Hinge Loss and Squared Hinge Loss, but these are less popular functions, and they require the class labels to be in the form [-1,1] rather than [0,1], which went against the pre-processing steps.

5.4 Bespoke Irish Model

This section details the implementation of the Bespoke Irish Model facet of the Problem Statement (see §1.1). It was required to select the highest performing model from the previous sections to tailor for Irish use. The highest performing model was found to be BERT (see Results 6 and so this model was selected. The implementation is congruent with the Approach (see §4), where the most powerful hate words were found from the original dataset, a new dictionary of Irish hate lexicon and benign words were synthesised for replacement.

5.4.1 Identifying Significant Hate Words

In order to identify the most potent hate words that would affect a prediction, it was decided to avail of one of the most interpretable models used within this research. The model was the Logistic Regression Classifier trained with TF-IDF features. The decision was partly because TF-IDF weighs the most significant words contributing to a particular class, meaning it was a more prudent choice relative to Word2Vec or BERT features. After all, they represent text in a more nuanced fashion and capture other text characteristics, like semantic relationships and context, rather than the presence of significant words. Moreover, choosing the Logistic Regression Classifier was a straightforward decision, given that it outperformed the other traditional classifiers (see Results section §6) and that it is a very interpretable classifier. Indeed, the model assigns weighted co-efficients to each feature supplied by TF-IDF (i.e. individual words with a TF-IDF score) that represent how significant the Classifier deems a feature in creating a probability. The model co-efficients were ranked from highest to lowest; identifying the most significant words in the model decision.

5.4.2 Choosing Irish Hate Words

In conjunction with identifying the hate words, the next step was creating a dictionary of Irish hate lexicon. The dictionary was derived from a list of hate words found in the Racial Slur Database[7]. The list included 50 hate-words and phrases. However, it was apparent that some of the hate words were more appropriate candidates than others. For example, the word "Clown" was in the Racial Slur Database for Ireland, but "Clown" is a universally hateful term when used in specific contexts, not just in Ireland. For this reason, Irish hate words were hand-picked to create the dictionary.

When choosing the words, a consideration was to ensure that the hate term would only make sense when used as a noun. For example, when choosing the significant hate words to replace, words like *"shit"* were not included because they can also be used to describe something. Likewise, the same premise held when selecting hateful Irish words from the Racial Slur Database.

Mapping Significant Hate Words to Irish Hate Words

Using the aforementioned choosing mechanism, the most significant hate words were mapped to the chosen list of Irish hate words. Choosing what Irish hate words would be mapped to each significant hate word was an arbitrary decision. Indeed, there was no method for weighing the significance of each Irish hate term like the hate words from the original dataset were. Since the identified hate words and the Irish hate words were all nouns, each Irish hate word was mapped to a significant hate word in no particular order. Another consideration was to analyse each word to separate them into different categories. For example, the word "nigga" has racial underpinnings, and "leprechaun" could be used as a racist term in some contexts. However, this often depends on the contexts and intent of these words and mapping them based on these characteristics was deemed imprudent and hence the first method was preferred. The mappings can be seen below:

- *"Bitch" => "Leprechaun"*
- *"Pussy" => "Proddy"*
- *"Hoe" =>"Spudnigger"*
- *"Idiot" =>"Fenian"*

[7] http://www.rsdb.org/race/irish Last accessed: 06/05/2022

- *"Faggot" =>"Nina"*
- *"Asshole" =>"Provo"*
- *"Cunt" =>"Shant"*
- *"Nigga" =>"Hibe"*

Choosing Benign Words

The choice of benign words to use for replacement was ultimately unimportant as long as they were uncontroversially benign. For example, the word "table" is not in any situation hateful. In order to determine this, two criteria were used to ensure the word was genuinely benign: 1) low occurrence rate within hate samples. 2) Low co-occurrence rate with the significant hate words. If the candidate benign word has virtually no occurrence within hate samples or co-occurrence with the significant hate words, the words are truly benign. These candidate words are assessed under these criteria within the results section (see §6). For consistency, the same number of benign words were chosen as the number of Irish hate words used for replacement, and the chosen words were all nouns, similar to the identified hate words and Irish hate words.

Mapping Significant Hate Words to Benign Words

Following the same principles as above, each benign word was mapped to an identified hate word. The mappings can be seen below:

- *"Bitch" => "Table"*
- *"Pussy" => "Sandwich"*
- *"Hoe" =>"Box"*
- *"Idiot" =>"Pencil"*
- *"Faggot" =>"Wallet"*
- *"Asshole" =>"Laptop"*
- *"Cunt" =>"Salad"*
- *"Nigga" =>"Bracelet"*

5.4.3 Replacing Hate Words

Following the approach, the identified hate words present within the Combined dataset were replaced at different replacement-rates. The dictionary mappings of significant hate words and Irish hate words were programmatically implemented in Python. An algorithm was created to take the dataset and a replacement-rate parameter, λ, that determines the rate of which a hate word is replaced with an Irish mapping. The parameter ranged from $\lambda = 1.0$, 0.75, 0.5 and 0.25. For example, a $\lambda = 0.75$ would indicate that "Bitch" is replaced with "Leprechaun" 75 % of the time. To enforce this, a random number would be generated between 1 and 100, and if the number were in the range of 0-λ then the word would be replaced, else it would remain. The algorithm generated four renditions of the dataset for each of the different replacement-rates. Both the training and validation sets were replaced for the Irish hate words, whereas only the validation sets were replaced with benign words.

5.4.4 Re-Training BERT

Utilising the new datasets, new BERT models were trained using the same architecture described in 5.3. This created 4 new BERT models trained with different numbers of hate word replacements. Each model was used to predict on the validation sets in order to produce the results presented in the Results Section (see §6).

5.4.5 Implementation Summary

Adhering to the Project Approach, the implementation carried out the following tangible and programmatic tasks that were used in this research:

1. **Datasets**
    - Collated the cross-platform Twitter and Wikipedia datasets and relabelled the data for the binary classification task ('hateful' or 'not hateful').
    - Created pre-processing pipelines to clean the data for optimal use in classification.
    - Split the datasets into train/test splits for validating model performances.
2. **Feature Engineering**
    - Implemented Feature Engineering techniques relating to the increasing level of success found within the related work, including TF-IDF, Word2Vec and BERT.
3. **Classifiers**
    - Implemented and fine-tuned a combination of traditional and Deep Learning classifiers, including Logistic Regression, Naïve Bayes, SVM and BERT.
    - Trained the classifiers with the the aforementioned feature sets.
4. **Bespoke Irish Model**
    - Identified the most significant words present within the original dataset by ranking the weights assigned to each word via the LR TF-IDF model.
    - Curated a set of Irish hate words and benign words.
    - Replaced the significant hate words with the Irish hate words at variable replacement-rates in order to re-train the best performing model.
    - Replaced the significant hate words with the benign ones at variable replacement rates only within the test set to evaluate the importance of hate words within the newly trained Irish model.

Chapter 6: Experimental Results and Evaluation

This chapter presents the results and evaluation of the research. Adhering to the Project Approach, the results and evaluation are conducted in accordance with the three main goals of the research. The goal of using modern techniques was evaluated by comparing the performance of the classifiers on the three platforms of data: Twitter, Wikipedia and Combined. Selecting the highest performing model from the previous evaluation, the goal of model generality was evaluated by using the three models generated by the approach (for example BERT trained with Twitter, Wikipedia and Combined data) and comparing how each of them performed on the different test sets. Lastly, the goal of tailoring the model for Irish use was evaluated by selecting the model with the greatest performance and comparing the model's performance re-trained with different hate word replacement rates on an Irish test set. It was also evaluated by comparing the performance of the final Irish model on a test set of variable hate word replacement with benign words to evaluate how important the model utilises hate words in its decision-making process.

The goals are not necessarily evaluated in the same order of their implementation, as many of the evaluation steps overlap. For example, to evaluate the goal of model generalisation, the goal of using modern techniques must also be evaluated.

6.1 Evaluation Metrics

This section outlines and justifies the evaluation metrics used when evaluating the model performances.

6.1.1 F1 Score

The Online Hate Detection techniques leveraged in this research are evaluated with the F1 score metric on the test datasets. The F1 Score is an evaluation metric that presents the combined information about precision and recall. The formula for the F1 Score can be seen below:

$$F1 = 2 * \frac{p * r}{p + r} \tag{6.1}$$

The precision refers to the ratio of correct predictions for the positive class, relative to the total number of positive predictions by the model. For example, if there are 100 samples and 90 of them are 'hateful' and the model predicts they are all hateful, the precision will be 90%. On the other hand, recall refers to the model's ability to guess the positive class correctly without mislabeling them as negative. The F1 Score is the harmonic balance between these two metrics.

It is prudent to use the F1 score metric in this research rather than other metrics because of the inherent class imbalances in the datasets. For example, using the accuracy metric in imbalanced

classification would be imprudent because a model can be engineered only to predict the majority class and would yield high accuracy scores. However, the F1 metric would be poor because it reflects the model's performance in predicting each class individually.

6.2 Modern Techniques

In order to evaluate the modern techniques each of the techniques highlighted in the Implementation (see §5) were used to predict against the each of the test sets for each platform.

6.2.1 Comparing Model Performance

Table 6.1 below depicts the F1 score performance for each classifier with the highest score in each platform highlighted in bold:

Table 6.1: F1 Scores - All Platforms

Classifier	Twitter	Wikipedia	Combined
LR TF-IDF	**0.953**	0.74	0.868
LR Word2Vec	0.907	0.684	0.549
NB TF-IDF	0.948	0.371	0.560
NB Word2Vec	0.444	0.561	0.530
SVM TF-IDF	0.949	0.732	0.862
SVM Word2Vec	0.901	0.673	0.552
BERT	0.945	**0.866**	**0.917**

Classifier Specific Analysis

Comparing the F1 Scores of all of the classification techniques, it was evident that BERT consistently outperformed all rivalling classifiers. Indeed, BERT achieved the highest F1 Score in two of the three datasets (Wikipedia and Combined), but was marginally outperformed by LR TF-IDF in the Twitter dataset. Unsurprisingly, BERT was the highest performing classifier, given its nuanced and complex deep learning architecture; it is unrivalled in its ability to interpret semantic and contextual patterns from the datasets compared to the other techniques. While the other classifiers performed significantly lower, LR TF-IDF and SVM TF-IDF performed very similarly, being within 0.01% of each other in all platforms and consistently outperformed the other classifiers with very successful results. NB was consistently the poorest performing classifier, ranking last in every platform regardless of the feature set.

Platform Specific Analysis

On average, the Twitter platform yielded the best F1 scores by a considerable degree for every classifier aside from NB Word2Vec. The results imply that the characteristics present within the Twitter dataset are more conducive for a model to identify hateful patterns within the text. It was discovered that the Twitter data is far more concise than the Wikipedia data (see §5.1),

corroborating that text length is an important factor in Online Hate Detection, where the longer the sample text is, the more difficult it is to identify hate. Indeed, the models didn't perform as well on the Wikipedia set as it contains more syntactically and semantically complex data to the more concise and to-the-point Twitter data. This is unsurprising given Wikipedia users are likely to take part in discussions, whereas Twitter is grossly limited in discussion by their short character limit. This is further ratified in that BERT performed very similarly to the classifiers with TF-IDF features in the Twitter dataset, but vastly outperformed them in Wikipedia – this conveys that context and semantics are not a critical factor in determining hate within the Twitter data, but are not as an important role in the Wikipedia data. In general, the Combined Score was a balance between the model's Twitter and Wikipedia score, implying that most of the models were successfully able to generalise the patterns within both datasets to perform well with the Combined test set, apart from the models trained with Word2Vec features.

Feature Specific Analysis

The results indicate that BERT features were the most efficacious in representing hate, given that it, in part, provided the best performance. This implies that subtle nuances in semantics and context are significant patterns to recognise within a model to perform well across multiple categories. Indeed, TF-IDF features are wholly contingent upon the significance of keywords, which is why it performed significantly better within the Twitter dataset relative to the Wikipedia dataset. Furthermore, due to the definition of "Online Hate" adopted in this research being predicated on the presence of hate terms, the task was inherently a good fit for TF-IDF features, given that it measures the significance of a word to a particular class. Hence, the set of words that are only present within the hate class, like swear words, are an automatic determination of whether the sample is hateful. TF-IDF does this very well, so it managed to score so highly within the Twitter data that was previously shown not to incorporate more structured and complex language. TF-IDF saw a significant decrease in F1 score across all classifiers in the Wikipedia dataset for this reason – Wikipedia data is more likely to contain samples containing hate not only due to the presence of a swear word, but due to information seeded within the semantics, context, tone and intent of the text. As such, the TF-IDF performance indicates that the presence of significant keywords is critical in determining hate within the Twitter dataset but is not as significant in datasets that have more complex and structured text. Word2Vec features categorically performed the poorest out of all techniques – even with balanced datasets that play more into the technique's strengths. The results show that Word2Vec struggled to derive the hateful patterns in the text with its vectorisation mechanism, and it conveys that this technique was unsuitable for the data used within this research.

6.2.2 Summary of Findings

The results from the Modern Techniques have provided some key insights to this research:

- BERT consistently outperforms all other techniques.
- TF-IDF features identify the significance of hate words and are successful for the Twitter platform but not as successful on the Wikipedia platform showing that the complexity and structure of text render TF-IDF features less effective.
- Word2Vec features were unable to capture the necessary information and patterns within the text data to make predictions across any of the datasets other than Twitter but still consistently performed the poorest.

The results enforce the evaluation of a vital goal of this research to ensure that the approach

uses modern techniques to generate state-of-the-art performance. Indeed, the results showed that BERT was the most effective Online Hate Detection technique, and it was selected as the technique to fine-tune and tailor for Irish use.

6.3 Generalization

Taking the best performing technique shown previously, its generalisation capabilities are measured by its performance on unseen data. Adhering to the approach, the three BERT models trained on each platform (Twitter, Wikipedia, Combined) are cross-validated with one another's test set.

6.3.1 Platform-Specific Performance

Table 6.2: F1 Scores - Cross-Platforms

Classifier	Twitter	Wikipedia	Combined
$\text{BERT}_{Twitter}$	**0.949**	0.745	0.865
BERT_{Wiki}	0.885	**0.866**	0.883
$\text{BERT}_{Combined}$	**0.949**	**0.886**	**0.917**

Single-Platform Model Analysis

Table 6.2 depicts the F1 scores of each BERT classifier (e.g. where $\text{BERT}_{Combined}$ refers to BERT trained with the Combined training set) on the different test sets, painting a picture to how well each model can perform on different kinds of unseen data. Unsurprisingly, the highest performing model for each platform was the models trained with that data platform. Furthermore, the models trained on only one platform performed significantly worse on the other platforms. The results indicate that models trained with only one platform of data lack the potency to generalise across other data platforms. For example, BERT trained with Twitter data performed the best with the Twitter test set but was vastly outperformed on the Wikipedia test set by the BERT model trained with Wikipedia data. Interestingly, the degree of falloff between test sets was most evident in the BERT model trained with Twitter data – it performed much more poorly on the Wikipedia test set than the Wikipedia model did on the Twitter test set. This is indicative that the model was able to interpret more general kinds of hate from the Wikipedia test set than the Twitter test set. It is reasoned that this was true due to Twitter's concise data samples and training with the Twitter data did not provide the model with enough complex semantics and structure to perform well with the more nuanced Wikipedia data. In contrast, the Wikipedia data provided enough patterns during training for the model to perform surprisingly well with the less complex Twitter data, demonstrating that the Wikipedia dataset was more robust for model generality than the Twitter dataset. Analysing the performance of the Wikipedia and Twitter models to the Combined dataset, Wikipedia had a slightly higher F1 score of 2%, further ratifying that the Wikipedia dataset was more potent in generalisation.

Cross-Platform Model Analysis

When observing the performance of the BERT model trained with the Combined dataset, it was evident that it had the best performance, retaining the efficacy of the single-platform training

whilst having the best results on the Combined test set. Indeed the Combined model matched the performance of the Twitter and Wikipedia models on their own test sets, implying that training the model with multiple sources of data will not affect its ability to learn patterns from the data. Moreover, the Combined dataset significantly outperformed the other models in performance for the Combined test set, showing that it is far more robust to unseen cross-platform data than the models trained with single-platform data.

6.3.2 Summary of Findings

The results from the Cross-Platform Performance have provided some key insights into the models:

- Wikipedia data provides more generalisation ability than Twitter data.
- Models trained with multiple data platforms will not affect their predictive capacity on single-platform data.
- Models trained with multiple data platforms generalise with unseen data far better than models trained with single-platform data.

The third item of the evaluation enforces a key goal of this research to create an approach that is generalised – able to perform well with different kinds of unseen data. The Cross-Platform Model Analysis results demonstrated this goal.

6.4 Bespoke Irish Model

This section discusses the results and analysis of the third goal issued in the Problem Statement (see §1.1). The goal is to tailor the highest performing model in the previous section for use within the Republic of Ireland. The highest performing model from the previous section was deduced to be the BERT model trained with the Combined dataset.

6.4.1 Most Significant Hate Words

Adhering to the project approach, the first facet in tailoring the model for Irish use was identifying the most significant hate words present within the dataset. The second highest performing model from the Modern Techniques section (see 6.1) was LR with TF-IDF features and its coefficients were ranked to depict the top ten most significant words that contribute toward the hateful prediction. Table 6.3 shows the top ten hate words with their decision co-efficients can be seen below:

Table 6.3: Top 10 Most Significant Hate Words

Feature	Importance
Bitch	1.8e9
Fuck	7.9e8
Pussy	2.9e5
Hoe	2.9e5
Idiot	1.9e5
Shit	1.7e5
Faggot	1.1e5
Asshole	5.5e4
Bullshit	3.0e4
Stupid	2.3e4

The Importance column depicts the values assigned to each word by the LR TF-IDF model – the higher the value, the more important the word has on the model decision. From the LR TF-IDF co-efficients, it is apparent that "Bitch" and "Fuck" were the top two most significant words for determining hate by a considerable margin, with their model co-efficient scores being orders of magnitudes above the other words. Unsurprisingly, the top hate words are all cuss, racial, prejudicial and derogatory words, indicating that the presence of these words was critical in determining hate for the LR TF-IDF model.

Moreover, the top 10 words encompassed a mix of nouns (e.g. "Bitch") and words that depend on their context to decide whether they are nouns or adjectives (e.g. "Shit"). For this research, it was required to select nouns for hate word replacement. As such, Table 6.4 depicts the top ten most significant hate nouns identified by LR TF-IDF.

Table 6.4: Top 10 Most Significant Hate Nouns

Feature	Importance
Bitch	1.8e9
Pussy	2.9e5
Hoe	2.9e5
Idiot	1.9e5
Faggot	1.1e5
Asshole	5.5e4
Cunt	8.0e3
Nigga	5.0e3
Nigger	3.8e3
Dick	3.2.0e3

Similarly, the top 10 hate nouns pertain to the same derogatory, prejudicial and racist nature as the top 10 hate words. The results of this were then used to map to the Irish hate words derived in the Implementation section (see §5). A total of 8 Irish hate words were used for this research and thus were mapped to the top 8 hate words, producing the following mappings:

- *"Bitch" => "Leprechaun"*
- *"Pussy" => "Proddy"*

- *"Hoe" => "Spudnigger"*
- *"Idiot" => "Fenian"*
- *"Faggot" => "Nina"*
- *"Asshole" => "Provo"*
- *"Cunt" => "Shant"*
- *"Nigga" => "Hibe"*

6.4.2 Hate Word Replacement with Irish Hate

The top 8 hate words that were identified previously were replaced with the Irish hate words with varying replacement-rates. New BERT models were trained with the data replaced with the different rates to produce results showing the model performances on the original test set and the test set replaced with the Irish hate words.

6.4.3 Irish Model Performance

Table 6.5 below depicts the performance of the BERT classifiers trained with different replacement-rates on the original test set and the Irish replacement test set (for example, $\text{BERT}_{100\%}$ is the BERT model re-trained with a replacement-rate of 100%):

Table 6.5: F1 Scores - Replacing with Irish Hate Words

Classifier	Original Test Set	100% Irish Replacement Test Set
$\text{BERT}_{100\%}$	0.908	0.906
$\text{BERT}_{75\%}$	0.910	0.908
$\text{BERT}_{50\%}$	0.911	**0.909**
$\text{BERT}_{25\%}$	0.910	0.899
$\text{BERT}_{Original}$	**0.917**	0.892

Original Test Set Analysis

The results evidenced that the original BERT model outperformed the re-trained BERT models in the original test set. This is unsurprising given that significant hate words are being replaced with out of vocabulary hate words not present within the test set. However, what is surprising is that the decrease in performance is incredibly marginal; the original BERT model outperforms the others by less than 0.01%. Indeed, the original BERT model scored 0.917 with the original test set, relative to the lowest performing replacement-rate of 100% scored 0.908. These results indicate that significant hate words are not a critical factor in determining hate, as completely removing the top 8 most significant hate words from the test set did not even reduce the F1 Score by a percent. Despite the decrease in performance is marginal, the trend is clear. Figure 6.1 illustrates the trends of the performance as the replacement-rate increases:

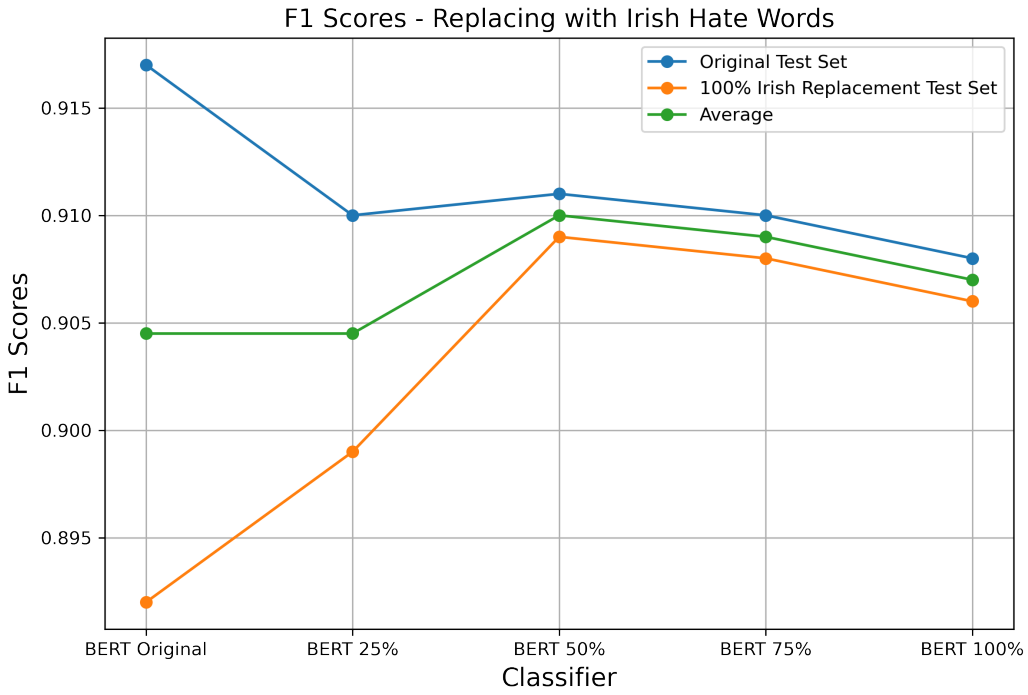

Figure 6.1: F1 Scores - Comparing Model Performance on the Original and 100% Irish Replacement Test Set

The blue line represents the model performances on the original test set. The results depict a general linear decline in performance as more hate words are replaced with Irish hate. There is a slight peak at 50% replacement, but there will be some variance in the model performance due to the small number of epochs used to train the data due to this project's time and resource constraints. The linear trend would likely be more apparent if the models could have been trained with more epochs to reduce variance.

As such, it was apparent that at the rate of which the significant hate words were replaced with Irish hate words, the performance on the original test set decreased. However, it was surprising that the decrease in performance was incredibly small, showing BERT's robustness and ability to discern hate from other mediums other than the presence of hate words.

Irish Test Set Analysis

For the Irish test set, the best performance was derived at a replacement rate of 50%. The orange curve illustrated by 6.1 shows a very apparent linear increase of performance in the Irish test set as more Irish vernacular is introduced to the model, but what is surprising is that this begins to decrease after 50%. Intuitively, this would continue to increase, but the results indicate that a replacement rate of 50% provides the best performance. Indeed, this could be due to variance of the model performances as it is natural to have slight fluctuations in performance between epochs. Only 10 epochs were used could partly be the reason. Moreover, the performance varies very slightly, approximately 1.5% between all of the models in this set, showing that the introduction of Irish vernacular into the training set only provides small performance increases when faced with Irish test data. The original BERT model performed the worst within the Irish test set, given that it was not trained with the abundance of Irish vernacular. Interestingly, the degree of performance increase when introducing Irish vernacular is much greater than the decrease of performance when removing significant hate words. This is illustrated by the green curve of the graph illustrating the average performance of the models between the two test sets. This provides exciting insight

that adding Irish vernacular aids the model's ability to perform with geographically specialised hate while still retaining the potency to detect general hate.

As such, this research deduced that a replacement-rate of 50% incurred the best average performance between the original test set and the Irish test set.

6.4.4 Benign Words

In similar fashion to the Irish hate word replacement, benign words were chosen to replace the top 8 significant hate words. The criteria used to choose hate words were outlined in the Project Approach (see §4). The 8 chosen benign words were evaluated to see how often they occur within samples labelled as hateful and evaluated to see how often they co-occur with any of the 8 most significant hate words. Table 6.6 below depicts the occurrence and co-occurrence rate of each of the benign words:

Table 6.6: Occurrence & Co-Occurrence Scores for each Benign Word

Benign Word	Occurrence Rate	Co-Occurrence Rate
Table	0.05%	0.00%
Sandwich	0.02%	0.00%
Box	0.12%	0.00%
Pencil	0.02%	0.00%
Wallet	0.02%	0.00%
Laptop	0.01%	0.00%
Salad	0.02%	0.00%
Bracelet	0.002%	0.00%

The results stipulate with the intuitively benign nature of the words, showing that all candidate words have a 0.00% co-occurrence rate with any of the top 8 significant hate words, meaning that none of the benign words occurs within the same samples as the most significant hate words. Moreover, the occurrence rate of the benign words even being within a hateful sample at all is incredibly low, with the highest word being "Box" with an occurrence rate of 0.12%, indicating that the word only occurs in 1 in every 1000 samples.

With these statistics the candidate benign words were deemed worthy to be used within this study. The words were mapped to the top 8 most significant hate words in no particular to produce the following mappings for replacement:

- *"Bitch" => "Table"*
- *"Pussy" => "Sandwich"*
- *"Hoe" =>"Box"*
- *"Idiot" =>"Pencil"*
- *"Faggot" =>"Wallet"*
- *"Asshole" =>"Laptop"*
- *"Cunt" =>"Salad"*

- *"Nigga" => "Bracelet"*

6.4.5 Hate Word Replacement with Benign Words

The top 8 most significant hate words within the Combined test set were replaced with the chosen benign words at different replacement-rates so that the highest performing model from the Irish hate word removal section could predict on it. This provides insight into how critical the model values the most significant hate words within the test set to perform well.

Figure 6.2 below shows the performance of the BERT model trained with 50% replacement with Irish hate words in the test sets of varying benign word replacement-rates. For illustrative purposes, this was also done with the LR TF-IDF model that has already shown to be heavily contingent upon the presence of the most important hate words to make a prediction.

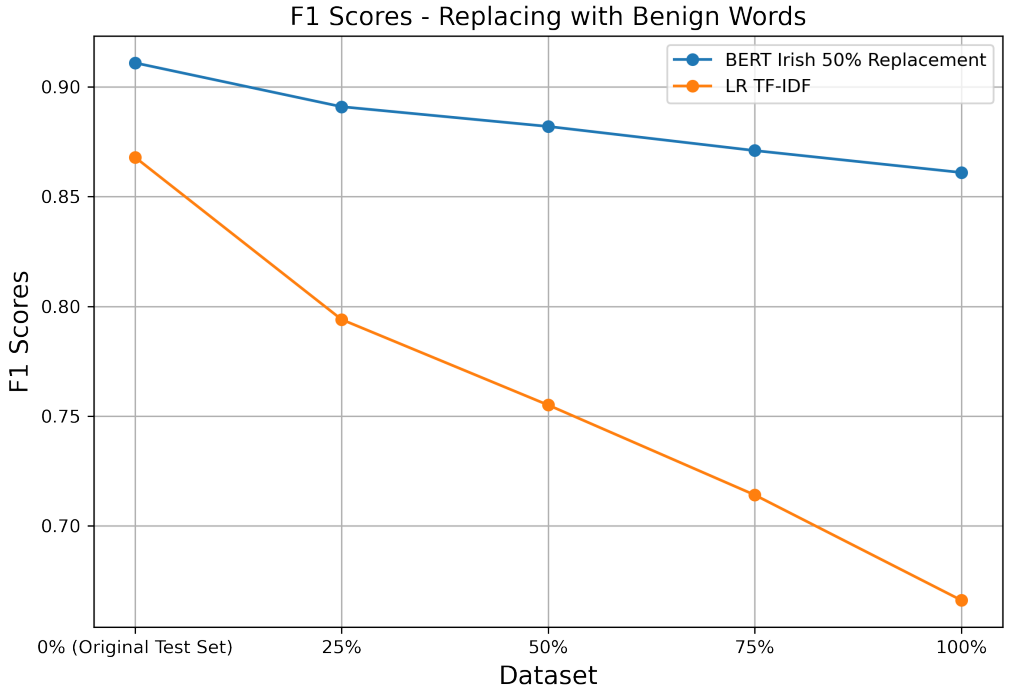

Figure 6.2: F1 Scores - Using BERT 50% Irish Model to Predict on Datasets with Varying Amounts of Hate Word Replacement with Benign Words

The results show that there is a clear linear relationship between the increase of benign word replacement and the decrease in performance. This is understandable given that the definition of Online Hate Detection includes the presence of hate words. Comparing the relationship of both the BERT model and the LR TF-IDF model, it is evident that the BERT model is far more robust to the hate word replacement than the LR TF-IDF model that plummets far more significantly. Even with 100% significant hate word replacement the Irish BERT model still produces reasonable results with over a 0.85 F1 Score.

Hence, despite the replacement being an important factor in the decrease in performance, BERT's resilience compared to LR TF-IDF shows its non-dependence on hate words and conveys it can determine hate from other patterns and information within the text. This corroborates the previous evaluation that introducing Irish hate vernacular improves its performance with Irish data, but it only improves marginally as BERT does not weigh the presence of individual words that heavily

when determining hate.

6.4.6 Summary of Findings

The results from the Modern Techniques have provided some key insights to this research:

- An ideal hate word replacement-rate with Irish vernacular was found to be 50% which generated the best average performance between the original test set and the Irish test set.
- Replacing significant hate words with Irish hate words increases performance within the Irish test set far more significantly than reducing performance within the original test set.
- The original BERT model performed the worst when faced with Irish data, whereas the adjusted Irish BERT models all performed well within the original test set.
- Replacing hate words with benign words affects the performance of the BERT model. However, BERT was far more resilient than the LR TF-IDF model, showing that significant hate words are not critical in determining hate.

The results enforce the final goal of this research, to tailor the highest performing model for Irish use. The results show that the model with 50% significant hate word replacement with Irish hate words performs the best on the Irish test set, whilst still retaining the potency on the original test set, showing that the Irish model is more optimised than the original model in an Irish context.

Chapter 7: **Summary and Conclusion**

This research showcased an approach in order to accomplish three main goals in combating "Online Hate":

1. Ensuring the approach is *Generalised*, meaning that it can perform well on multiple platforms of data.
2. Ensuring the approach is *Modern*, meaning that it leverages state-of-the-art Online Hate Detection techniques to maximise performance.
3. Ensuring the approach is *Bespoke For Irish Use* – tailored and optimised for use in Ireland.

This research experimented with multiple modern Online Hate Detection techniques, with BERT consistently outperforming other traditional classifiers by a significant degree. Moreover, incorporating multi-platform Twitter and Wikipedia data demonstrated BERT to be generalised in practice, with the BERT model trained with a combination of both data vastly outperforming the BERT models trained on either platform independently. Lastly, this research tailored the highest performing BERT model for Ireland by replacing the top 8 most significant hate words in the training set with Irish hate words. Doing so demonstrated performance increases when faced with data comprised of Irish hate words relative to the baseline BERT model.

Some key findings include:

- BERT significantly outperformed traditional classifiers in virtually all datasets – it was only marginally outperformed within the Twitter dataset by LR TF-IDF.
- The presence of significant hate words are not critical for prediction within the BERT model.
- BERT trained with cross-platform data retained each platform's performance and demonstrated its ability to generalise within the Combined test set.
- Replacing the top 8 most significant hate words with Irish hate words within the data yielded the best performance results with a 50% replacement rate.
- Increasing Irish vernacular within the BERT model increases performance with Irish data far more than decreasing performance with the original data.
- Replacing significant hate words with benign words showed that the tailored BERT model could better determine hate from other text features relative to other high-performing classifiers like LR TF-IDF.

Limitations and Future Work

The project's approach has shown slight model improvement when optimised for Irish use. However, this does not mean that it comes without limitations. Firstly, thorough hyper parameter tuning was not a viable tool due to BERT's uncanny training time. Hyper parameters like learning rate,

optimiser and loss function were chosen with intuition and success from related work rather than experimental inference. Indeed, it could be possible that a greater performance could be achieved with scrupulous parameter tuning. As such, introducing high power GPUs to kerb BERT's training time could offer performance increases in future research.

Secondly, the approach was contingent upon significant hate word replacement, but BERT does not weigh significant words crucially in its decision – limiting its performance gains on the Irish training sets. The results showed this as the BERT model only gained approximately 1.5% F1 Score with the introduction of Irish vernacular. Thus, it would be prudent to explore other ideas to introduce Irish flavours to the text data other than word replacement. BERT has been shown to deduce hate from context, structure and semantics, so further research into how this text features change in different geographies could provide exciting work in the future.

Lastly, the approach is limited as it is a binary classification problem. Related work has placed value in discerning targeted groups from hateful text [49], for example understanding whether the hate was based on race, sexism or general toxicity. As such, this offers exciting potential for future work that could engineer a model that can discern Irish forms of hate from other taxonomies of hate.

Acknowledgements

This research could not have been completed if not for the inspiration and patience of the project supervisor Simon Caton, for both guidance and conceptualising the research idea. Family and friends were also essential supporting factors throughout this research.

Bibliography

1. Duggan, M. Online harassment 2017 (2017).
2. Calvete, E., Orue, I. & Gámez-Guadix, M. Cyberbullying victimization and depression in adolescents: The mediating role of body image and cognitive schemas in a one-year prospective study. *European Journal on Criminal Policy and Research* **22,** 271–284 (2016).
3. Burnap, P. & Williams, M. L. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & internet* **7,** 223–242 (2015).
4. Matamoros-Fernández, A. & Farkas, J. Racism, Hate Speech, and Social Media: A Systematic Review and Critique. *Television & New Media* **22,** 205–224 (2021).
5. Salminen, J. *et al.* Developing an online hate classifier for multiple social media platforms. *Human-centric Computing and Information Sciences* **10,** 1–34 (2020).
6. Falkum, I. L. The how and why of polysemy: A pragmatic account. *Lingua* **157,** 83–99 (2015).
7. Naseem, U., Razzak, I. & Hameed, I. A. Deep Context-Aware Embedding for Abusive and Hate Speech detection on Twitter. *Aust. J. Intell. Inf. Process. Syst.* **15,** 69–76 (2019).
8. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y. & Chang, Y. *Abusive language detection in online user content* in *Proceedings of the 25th international conference on world wide web* (2016), 145–153.
9. Zhang, Z., Robinson, D. & Tepper, J. *Detecting hate speech on twitter using a convolution-gru based deep neural network* in *European semantic web conference* (2018), 745–760.
10. Mondal, M., Silva, L. A. & Benevenuto, F. *A measurement study of hate speech in social media* in *Proceedings of the 28th ACM conference on hypertext and social media* (2017), 85–94.
11. Gagliardone, I., Gal, D., Alves, T. & Martinez, G. *Countering online hate speech* (Unesco Publishing, 2015).
12. Kocoń, J. *et al.* Offensive, aggressive, and hate speech analysis: From data-centric to human-centered approach. *Information Processing & Management* **58,** 102643 (2021).
13. West-Newman, C. L. Reading hate speech from the bottom in Aotearoa: Subjectivity, empathy, cultural difference. *Waikato L. Rev.* **9,** 231 (2001).
14. Davidson, T., Warmsley, D., Macy, M. & Weber, I. *Automated hate speech detection and the problem of offensive language* in *Proceedings of the International AAAI Conference on Web and Social Media* **11** (2017).
15. Wulczyn, E., Thain, N. & Dixon, L. *Ex machina: Personal attacks seen at scale* in *Proceedings of the 26th international conference on world wide web* (2017), 1391–1399.
16. Wang, J., Nansel, T. R. & Iannotti, R. J. Cyber and traditional bullying: Differential association with depression. *Journal of adolescent health* **48,** 415–417 (2011).
17. Khurana, D., Koli, A., Khatter, K. & Singh, S. Natural language processing: State of the art, current trends and challenges. *arXiv preprint arXiv:1708.05148* (2017).
18. Gu, J. & Yu, Z. Data annealing for informal language understanding tasks. *arXiv preprint arXiv:2004.13833* (2020).
19. Vidgen, B., Thrush, T., Waseem, Z. & Kiela, D. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761* (2020).

20. Dixon, L., Li, J., Sorensen, J., Thain, N. & Vasserman, L. *Measuring and mitigating unintended bias in text classification* in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* (2018), 67–73.

21. Gröndahl, T., Pajola, L., Juuti, M., Conti, M. & Asokan, N. *All you need is" love" evading hate speech detection* in *Proceedings of the 11th ACM workshop on artificial intelligence and security* (2018), 2–12.

22. Gitari, N. D., Zuping, Z., Damien, H. & Long, J. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering* **10,** 215–230 (2015).

23. Salminen, J. *et al. Anatomy of online hate: developing a taxonomy and machine learning models for identifying and classifying hate in online news media* in *Twelfth International AAAI Conference on Web and Social Media* (2018).

24. Tang, D. *et al. 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)* (2014), 1555–1565.

25. Mou, L. *et al.* How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111* (2016).

26. Gambäck, B. & Sikdar, U. K. *Using convolutional neural networks to classify hate-speech* in *Proceedings of the first workshop on abusive language online* (2017), 85–90.

27. Saksesi, A. S., Nasrun, M. & Setianingsih, C. *Analysis Text of Hate Speech Detection Using Recurrent Neural Network* in *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)* (2018), 242–248.

28. Pitsilis, G. K., Ramampiaro, H. & Langseth, H. Effective hate-speech detection in Twitter data using recurrent neural networks. *Applied Intelligence* **48,** 4730–4742 (2018).

29. Wolf, T. *et al.* Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* (2019).

30. Liu, P., Li, W. & Zou, L. *NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers* in *Proceedings of the 13th international workshop on semantic evaluation* (2019), 87–91.

31. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

32. Gunning, D. *et al.* XAI—Explainable artificial intelligence. *Science Robotics* **4** (2019).

33. Pérez-Landa, G. I., Loyola-González, O. & Medina-Pérez, M. A. An Explainable Artificial Intelligence Model for Detecting Xenophobic Tweets. *Applied Sciences* **11,** 10801 (2021).

34. Wang, C. *Interpreting neural network hate speech classifiers* in *Proceedings of the 2nd workshop on abusive language online (ALW2)* (2018), 86–92.

35. Ousidhoum, N., Lin, Z., Zhang, H., Song, Y. & Yeung, D.-Y. Multilingual and multi-aspect hate speech analysis. *arXiv preprint arXiv:1908.11049* (2019).

36. Basile, V. *et al. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter* in *13th International Workshop on Semantic Evaluation* (2019), 54–63.

37. Hall, M., Mazarakis, A., Chorley, M. & Caton, S. *Editorial of the special issue on following user pathways: Key contributions and future directions in cross-platform social media research* 2018.

38. Aggarwal, P., Horsmann, T., Wojatzki, M. & Zesch, T. *LTL-UDE at SemEval-2019 Task 6: BERT and two-vote classification for categorizing offensiveness* in *Proceedings of the 13th International Workshop on Semantic Evaluation* (2019), 678–682.

39. Krouska, A., Troussas, C. & Virvou, M. *The effect of preprocessing techniques on Twitter sentiment analysis* in *2016 7th international conference on information, intelligence, systems & applications (IISA)* (2016), 1–5.

40. Al-Saqqa, S. & Awajan, A. *The use of word2vec model in sentiment analysis: A survey* in *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control* (2019), 39–43.

41. Etaiwi, W. & Naymat, G. The impact of applying different preprocessing steps on review spam detection. *Procedia computer science* **113,** 273–279 (2017).

42. Scott, S. & Matwin, S. *Feature engineering for text classification* in *ICML* **99** (1999), 379–388.

43. Lilleberg, J., Zhu, Y. & Zhang, Y. *Support vector machines and word2vec for text classification with semantic features* in *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)* (2015), 136–140.

44. Devlin, J. & Chang, M.-W. Open sourcing BERT: State-of-the-art pre-training for natural language processing. *Google AI Blog* **2** (2018).

45. Wongkar, M. & Angdresey, A. *Sentiment analysis using Naive Bayes Algorithm of the data crawler: Twitter* in *2019 Fourth International Conference on Informatics and Computing (ICIC)* (2019), 1–5.

46. Lin, M., Chen, Q. & Yan, S. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

47. Choi, D. *et al.* On empirical comparisons of optimizers for deep learning. *arXiv preprint arXiv:1910.05446* (2019).

48. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

49. Castelle, M. *The linguistic ideologies of deep abusive language classification* in *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)* (2018), 160–170.