2022

# Evaluating the Performance Impact of Fine-Tuning Optimization Strategies on Pre-Trained DistilBERT Models Towards Hate Speech Detection in Social Media

Aidan McGovern
*Technological University Dublin, Ireland*

## Recommended Citation

mcGowran, A. (2022). Evaluating the Performance Impact of Fine-Tuning Optimization Strategies on Pre-Trained DistilBERT Models Towards Hate Speech Detection in Social Media. [Technological University Dublin].

# Evaluating the Performance Impact of Fine-Tuning Optimization Strategies on Pre-Trained DistilBERT Models Towards Hate Speech Detection in Social Media

**Aidan McGowran**

A dissertation submitted in partial fulfilment of the requirements of
Technological University Dublin for the degree of
M.Sc. in Computer Science (Data Analytics)

**April 03, 2022**

# DECLARATION

I certify that this dissertation which I now submit for examination for the award of MSc in Computing Science (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

**Signed:** Aidan McGowan

**Date:** **03 April 2022**

# ABSTRACT

Hate speech can be defined as forms of expression that incite hatred or encourage violence towards a person or group based on race, religion, gender, or sexual orientation. Hate speech has gravitated towards social media as its primary platform, and its propagation represents profound risks to both the mental well-being and physical safety of targeted groups. Countermeasures to moderate hate speech face challenges due to the volumes of data generated in social media, leading companies, and the research community to evaluate methods to automate its detection. The emergence of BERT and other pre-trained transformer-based models for transfer learning in the Natural Language Processing (NLP) domain has enabled state-of-the-art performance in hate speech detection. Yet, there are concerns around the performance at scale and environmental costs of increasingly large models.

The DistilBERT model is a more compact, faster transformer-based architecture, which offers a more scalable and economical alternative to the BERT model it is distilled from. This research evaluates the performance of the pre-trained DistilBERT fine-tuned for hate speech classification, using a dataset of labelled data from Twitter and Gab, and compares it against the BERT equivalent. Furthermore, this study evaluates strategies to optimize fine-tuning of DistilBERT models to improve classification performance, including weight re-initialization, layer-wise learning rate decay, and intermediate transfer learning.

This research demonstrated that the combined application of layer-wise rate decay and weight re-initialization when fine-tuning a DistilBERT model resulted in an average macro F1 score improvement of 2.08% compared to a BERT base model. The findings recommend further investment by the research community into lighter weight models over the larger transformer equivalents due to the benefits of speed, scalability, costs, and environmental impact.

**Keywords:** *Text Classification, Transformers, Transfer Learning, Fine-tuning, Optimization, Hate Speech, BERT, DistilBERT*

# ACKNOWLEDGEMENTS

*I want to thank my project supervisor, Dr. Bojan Božic, for his guidance and feedback throughout this endeavour.*

*I would also like to thank all my lecturers at TU Dublin who helped contribute towards this research as their classes developed the fundamental skills needed to complete this thesis. In particular, I would like to specifically thank Dr. Luca Longo and Dr. John Gilligan for their training in turning an idea into a proposal and instilling academic rigour and discipline into the process.*

*Finally, I reserve a special thank you for my wonderful wife, Isabel McGowran. Aside from being temporarily a single parent of two boys during numerous COVID-19 lockdowns while I retreated into a shed to complete this course, it wouldn't be possible without her loving support, constant encouragement, and remarkable patience.*

# TABLE OF CONTENTS

# TABLE OF FIGURES

## TABLE OF TABLES

# 1 INTRODUCTION

Online social media is increasingly becoming part of modern life in society, with 2.95 billion active social media users worldwide in 2019 (Dwivedi et al., 2021). Social media offers a platform for people to communicate and connect, but these networks' popularity and reach inevitably lead to toxic misuse by members of these communities. Malicious actors gravitate towards online social networks to espouse offensive and hateful views due to the relative anonymity these platforms provide (Burnap & Williams, 2015). These views are categorized as hate speech when they disparage a person or group based on traits such as gender, race, ethnicity, or sexual orientation and can lead to inciting hatred towards targeted groups, culminating in violence against these groups (Mathew et al., 2019). The proliferation of hate speech is highlighted in global surveys performed by Hawdon et al. (2017), which estimated that up to 53% of users had been exposed to hate speech online.

Social media companies and governments have attempted to curb hate speech by introducing legislation and policies while balancing the dichotomy of supporting the freedom of speech and censorship of hate speech. Social media companies use moderators to help distinguish between what is hate speech and what is merely offensive but the sheer volume of user activity requires automated approaches to support hateful language detection.

This text classification challenge has received significant attention from the research community, which has addressed this problem by applying Natural Language Processing (NLP) and Machine Learning algorithms to classify hate speech (Fortuna & Nunes, 2018). The emergence of pre-trained language models as state-of-the-art in NLP research has seen these increasingly applied to the challenge of hate speech detection. Yet, concerns remain around these ever-growing language models' financial and environmental costs (Bender et al., 2021). This study will assess if smaller, more cost-effective models offer a viable alternative to the larger language models.

This chapter will provide further background into the definition of hate speech and text classification strategies, clarify the problem statement this research will address, and the objectives required to accomplish this. Furthermore, this chapter will give an overview of the methodologies used, the research project's scope and limitations, and will conclude with an outline of the remainder of this research document.

**Reader discretion advised. This research contains uncensored content extracted from research and datasets covering hateful and offensive language. Readers may find this content disturbing, offensive, or distasteful.**

## 1.1 Background

This research focuses on using Natural Language Processing (NLP) and Machine Learning for text-mining and classification. The NLP domain is a branch of Artificial Intelligence that researches how machines can understand human language. NLP provides techniques to transform unstructured language into structured normalized representations. These representations enable machine learning algorithms to gain insights into speech or textual data. NLP algorithms offer the tools to implement sentiment analysis, topic extraction, text summarization, entity recognition, and automated question answering. Machine Learning is a separate branch of Artificial Intelligence, which studies the development of systems that can automatically learn and improve without explicit programming, using algorithms to identify patterns in data to help generate predictions of outcomes. Advances in this field have seen the development of Neural Networks, inspired by the workings of the human brain, which enable deep learning of patterns and representations within data.

The structured output of NLP is used in conjunction with machine learning algorithms to perform text classification. NLP techniques extract features from data for use within machine learning classification algorithms. Transfer Learning models, which use neural networks pre-trained on large sets of labelled data for similar tasks, provide new opportunities for NLP problems, providing baseline models that can be fine-tuned to more specific tasks on smaller data sets. Performance gains in these specific tasks can be achieved by optimizing the fine-tuning stage by applying techniques designed to improve the learning of the specific task.

Hate speech in social media is an increasingly topical problem, with social media companies struggling to identify and remove harmful hate speech content due to the volume of data produced. Research has looked towards NLP and machine learning for automated hate speech detection to address this problem. This research will focus on these methods for hate speech detection in social media, leveraging datasets from prior research, which comprise posts to the Twitter and Gab social networks.

## 1.2 Research Problem

Hate speech can be classified as any public speech that expresses hatred towards or encourages violence towards an individual or group based on their race, gender, sexuality, religion, disability, nationality, or ethnic group. It is becoming an increasingly prevalent issue within society. Social media platforms are being used as the primary avenue to share discriminatory views and target groups. This hate speech can profoundly impact victims' mental health and safety, potentially leading to conflict and violence on a broader scale (Burnap & Williams, 2015).

The sheer volume of content produced on social media platforms requires automated solutions to detect hateful content (Ullmann & Tomalin, 2020). Automation introduces significant challenges, evidenced by social media companies' limited success in addressing hate speech. The subjective nature of hate speech, multilingual nuances, and evolving terminology each contribute to the complexity of the problem. Classification algorithms must be designed to protect minority groups in the community without impacting the freedom of speech.

The popularity of social media and the increased public debate around hate speech has seen an increase in focus on the subject by the computer science research community. Traditional approaches using NLP for feature extraction and machine learning algorithms provided solutions to automation, which suffered from a conflation of hateful language and language that is merely offensive. The inability to distinguish between content that violates hate speech policies and content that is offensive but acceptable is a significant limitation to a fully automated solution, as these misclassifications could contravene legal rights for the freedom of speech.

The emergence of Transformer based architectures as state of the art in the NLP domain has seen these models increasingly applied to address these previous limitations in hate speech detection. These pre-trained deep learning models, such as the ground-breaking BERT, provide a richer contextual understanding of languages that are important to address the nuances between hateful and offensive content. These models have contributed towards state-of-the-art results in hate speech detection but pose new challenges to their application in social media networks. These models' large size, slower training times, and inference times pose difficulties in scaling their training, deployment, and execution across the complex infrastructure needed to support the volume of usage on social media networks.

Recent research has resulted in the development of lighter weight models, including DistilBERT and Facebook's Linformer, to address notoriously slow training and deployment times of the larger Transformer neural networks (Wang et al., 2020), with the latter employed by Facebook and Instagram platforms to aid with hate speech detection. While these lighter models offer comparable performance with standard models, they are not state-of-the-art in performance and require a trade-off between model performance and speed. When applied to BERT, research has demonstrated performance gains through fine-tuning optimization strategies, such as weight re-initialization, layer-wise linear rate decay (LLRD), and intermediate task transfers (Zhang et al., 2020). These strategies could help reduce the performance trade-off margins, raising the following research question:

***"Can a smaller, faster transformer model such as DistilBERT outperform a standard BERT model in the task of hate speech classification of social media data through the application of weight re-initialization, layer-wise linear rate decay, and intermediate task transfers during fine-tuning?"***

This research will address this question by breaking it down into the below series of sub-questions for which experiments can be developed.

**Sub-Question A:** Does a standard BERT model outperform a DistilBERT model in the task of hate speech classification?

**Sub-Question B:** Does re-initializing the weights of layers in a DistilBERT model before fine-tuning improve its performance?

**Sub-Question C:** Does the application of LLRD in a DistilBERT model improve its performance?

**Sub-Question D:** Does fine-tuning a DistilBERT model on an intermediate task before fine-tuning on the hate speech classification task improve its performance?

**Sub-Question E:** Which combination of weight re-initialization, LLRD, and intermediate task transfers with a DistilBERT model results in the best performance?

**Sub-Question F:** Do any of the fine-tuning optimization strategies result in a DistilBERT model that significantly outperforms a standard BERT model in the task of hate speech classification of social media data?

## 1.3 Research Objectives

The primary objective of this research is to compare the performance in hate speech detection of a standard BERT model and DistilBERT models employing the weight re-initialization, layer-wise linear rate decay, and intermediate task transfer fine-tuning optimization strategies. The goal is to determine if the smaller, faster DistilBERT model can be optimized to outperform BERT and offer a quicker, cheaper, and more scalable solution to the problem of hate speech classification in social media. The null hypothesis was formulated to test this objective, which assumes that the standard BERT base model will outperform the optimized DistilBERT model.

**Null Hypothesis:** *The pre-trained BERT base model statistically outperforms a DistilBERT model, which employs fine-tuning optimization strategies, on the average macro F1 score when fine-tuned on the social media dataset for the target task of hate speech classification.*

**Alternate Hypothesis:** *A DistilBERT model optimized using a combination of weight re-initialization, LLRD, and intermediate task transfers, outperforms BERT in the* *macro F1 score for the classification of hate speech when fine-tuned on the social media dataset.*

To test this hypothesis, research objectives were defined, which correspond to the research sub-questions formulated in section 1.2.

**Objective A:** Fine-tune standard BERT base model and DistilBERT model on social media dataset to develop baselines for comparison in the performance of hate speech detection.

**Objective B:** Fine-tune DistilBERT model with an optimal number of re-initialized weights on the social media dataset.

**Objective C:** Fine-tune DistilBERT model with optimal layer-wise learning rate decay on the social media dataset.

**Objective D:** Fine-tune DistilBERT model on a questioning and answering task dataset before fine-tuning this model on the hate speech classification task with the social media dataset.

**Objective E:** Fine-tune DistilBERT models using combinations of weight re-initialization, LLRD, and intermediate task transfers.

**Objective F:** Compare and evaluate the performance of models generated in objectives A, B, C, D, and E using the average macro F1 score to determine the best performing model in the classification of hate speech on the social media dataset.

A few key additional tasks are listed below; these are essential prerequisites to completing the objectives.

- Comprehensive Literature review to understand the current state-of-the-art in hate speech detection, the performance of transformer models in this domain, and analysis of optimization strategies and configuration to improve performance.

- Collection of labelled social media data from previous literature that will provide a generalized balanced dataset for use in hate speech classification.
- Dataset exploration, analysis, and pre-processing in preparation for fine-tuning models.

## 1.4 Research Methodologies

This section will describe the methodologies used for this research, referencing the research onion proposed by Saunders et al. (2016). This research follows a *positivist* philosophy, using a *deductive* approach to formulate a hypothesis that can be tested through empirical research and the development of experiments. While this research deals with textual data, the study is *quantitative* as data is encoded into a numerical format so the deep learning classification models can process it.

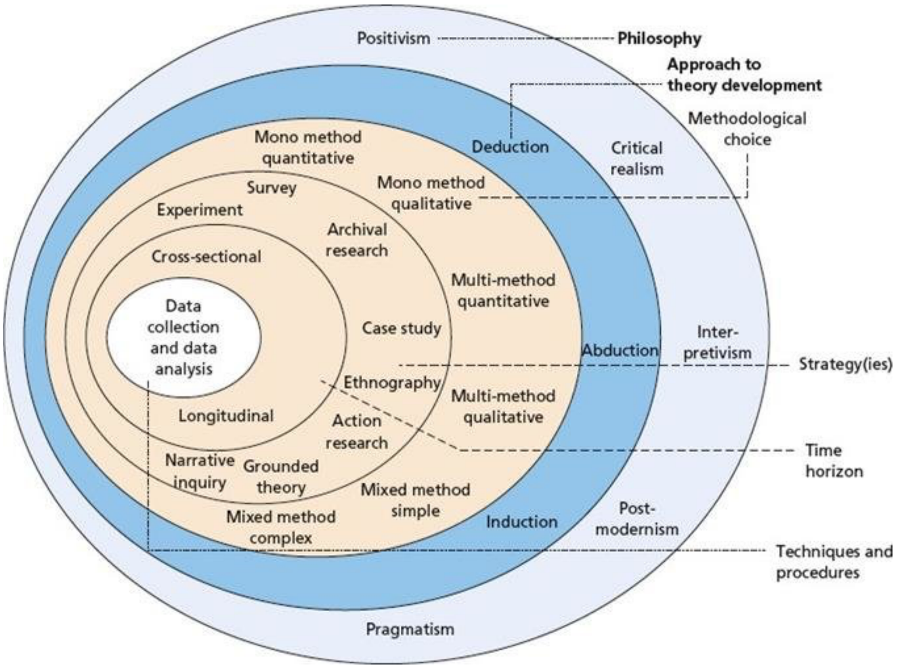


**Figure 1. 1: Saunders et al.'s Research Onion**

An *experimental strategy* is followed in this research, defining the experiments to test the research questions and objectives posed to refute the null hypothesis. The *mono method* is the choice for this research because the datasets leveraged will be encoded solely into quantitative data when testing the hypothesis. The datasets are *secondary* datasets derived from prior research. While the social media posts within the datasets span a range of time, the time horizons will be treated as *cross-sectional* for this research.

This research has employed the CRISP-DM (Wirth, 2000) framework to structure the research process, which is reflected in the structure of this document. The 'Literature Review' covered in Chapter 2 reflects the *Business Understanding* phase. The *Data Understanding* and *Data Preparation* phases are covered in the 'Design and Methodology' section in Chapter 3. The *Data Modelling* and *Model Evaluation* phases are represented in Chapter 4 on 'Results, Evaluation and Discussion.' The cyclical nature of the CRISP-DM model, with a feedback loop contributing to model evolution and increased business understanding, is reflected in the Chapter 5 sections on 'Contributions and Impact' and 'Future Work and Recommendations.'

## 1.5 Scope and Limitations

This research aims to verify if optimized DistilBERT models can offer lighter weight and scalable alternatives to larger models such as BERT in the classification of hate speech detection in social media. The research focuses on three specific optimization strategies; weight re-initialization, layer-wise linear rate decay, and intermediate task transfers. This is by no means an exhaustive list of optimization strategies. Techniques such as dropout and mixout regularization can impact performance but are out of the scope of this research.

The research focuses solely on the text content within social media posts for the features used during classification. Other factors such as the history of the offending user, the demographics of targeted users, or hypermedia such as images, video, or links could improve hate speech detection but are not assessed in this research. Furthermore, this research will focus solely on English language models; therefore, any findings may not apply to multi-language classification problems.

The structure of social media posts varies from standard English as character restrictions have resulted in shorter sentences and the use of internet slang and abbreviations. This unique language model could prevent any findings within this research from being applied to general text classification problems.

Limitations regarding access to social media APIs and the expense associated with the annotation of large datasets were prohibitive to curating a new dataset; therefore, this research leverages datasets curated through prior research. These datasets are combined to balance the dataset and improve the model's generalization, preventing this research's performance metrics from being compared to previous research referencing these datasets.

This research uses the BERT standard model as a baseline for comparison. Larger variations of BERT models offer superior performance than the standard model but were too computationally expensive to model due to the limited resources available. Similarly, hyperparameter searches and optimization strategies were not applied to the BERT model due to resource and time constraints. The assumption is that these may improve the classification performance of BERT. As this research is primarily focused on increasing the performance of the DistilBERT models through optimization strategies, these were deemed to be justifiable omissions.

## 1.6 Document Outline

The remainder of this dissertation comprises four further chapters, which are organized as follows:

- **Chapter 2 – Literature Review:** This chapter provides an overview of social studies conducted on the definition and impact of hate speech. This is followed by a comprehensive review of research into the problem of hate speech detection, the datasets available, NLP feature modelling, and the various machine learning algorithms used for classification. This will be followed by a review of the emergence of Transformer architectures as the state-of-the-art in the NLP domain and their application towards hate speech classification problems, concluding with the research gaps and opportunities.
- **Chapter 3 – Design and Methodology**: This chapter provides an overview of the design methodologies and technologies used to conduct the experiments in this research. Details will be provided on the datasets used for training and evaluating models, including content exploration and the strategies used for data preparation. This chapter will conclude with a design summary of the

experiments conducted, including the BERT and DistilBERT models used, the optimization strategies employed, and how performance will be evaluated.

- **Chapter 4 – Results, Evaluation, and Discussion:** This chapter will provide details of the implementation of the data preparation and experiments outlined in Chapter 3. The results from each experiment will be presented, and the chapter concludes with a discussion on the findings and evidence to support or reject the null hypothesis.
- **Chapter 5 – Conclusion**: This chapter will summarize the research project, including results and contributions to the body of research, along with recommendations and opportunities for further work based on this study.

# 2 LITERATURE REVIEW

This chapter provides a comprehensive overview of published research related to hate speech and its automatic detection. The chapter opens with a review of the definition of hate speech and its prevalence in social media. The chapter continues with an overview of the use and limitations of traditional text classification methodologies, leveraging NLP and machine learning, for hate speech classification in research. This will be followed by a review of the emergence of deep learning models as state of the art in NLP and hate speech detections domains, the challenges posed by increasing model sizes, and the research which seeks to address these. The chapter will conclude with the gaps in the literature and a summary of the findings.

**Search Keywords**: *hate*, *hate speech, hate crime, text classification, social media, twitter, gab, facebook, reddit, toxic, transformers, deep learning, fine-tuning*

## 2.1 Hate Speech in Social Media

Hate Speech is a challenging, complex, and subjective topic, which is reflected in the lack of an agreed-upon definition across academia, policymakers, and social media companies (Siegel, 2020). While a wealth of literature exists discussing the causes, impact, and potential solutions to address hate speech, little academic research has focused on systematically defining the term (Sellars, 2016). Similar themes can be extracted from the different definitions across research, policies, and social media codes of conduct listed in Table 2.1. In an attempt to summarize these views, hate speech can be broadly defined as language or content which can degrade, promote hatred or incite violence towards any individual or group based on race, gender, religion, sexual orientation, nationality, or ethnicity.

Justice Potter Stewarts' famous reflection of "I know it when I see it" regarding pornography in the Jacobellis v. Ohio trial (Gewirtz, 1996) cannot be applied to hate speech given the ambiguity posed by these different interpretations. These ambiguities inevitably lead to challenges in attempting to recognize hate speech due to the subjective nature of their interpretations, as noted in previous research which reviewed the low agreement among annotators when crowdsourcing the labelling of hateful content  (Fortuna & Nunes, 2018; Nobata et al., 2016; Davidson et al., 2017).

| **Source** | **Definition** |
|---|---|
| Siegel, 2020 | "hate speech is understood to be bias-motivated, hostile, and malicious language targeted at a person or group because of their actual or perceived innate characteristic" |
| Fortuna & Nunes, 2018 | “Hate speech is language that attacks or diminishes, that incites violence or hate against groups, based on specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual orientation, gender identity or other, and it can occur with different linguistic styles, even in subtle forms or when humour is used.” |
| Nobata et al., 2016 | “Language which attacks or demeans a group based on race, ethnic origin, religion, disability, gender, age, disability, or sexual orientation/gender identity.” |
| Irish Prohibition of Incitement to Hatred Act, 1989[1] | “hatred against a group of persons in the State or elsewhere on account of their race, colour, nationality, religion, ethnic or national origins, membership of the travelling community or sexual orientation” |
| EU Code of Conduct[2] | “all conduct publicly inciting to violence or hatred directed against a group of persons or a member of such a group defined by reference to race, colour, religion, descent or national or ethnic” |
| Facebook[3] | “We define hate speech as a direct attack against people – rather than concepts or institutions – on the basis of what we call protected characteristics: race, ethnicity, national origin, disability, religious affiliation, caste, sexual orientation, sex, gender identity and serious disease. We define attacks as violent or dehumanising speech, harmful stereotypes, statements of inferiority, expressions of contempt, disgust or dismissal, cursing and calls for exclusion or segregation” |
| Twitter[4] | **“Hateful conduct**: You may not promote violence against or directly attack or threaten other people on the basis of race, ethnicity, national origin, caste, sexual orientation, gender, gender identity, religious affiliation, age, disability, or serious disease. primary purpose is inciting harm towards others on the basis of these categories.” |

**Table 2.1:  Hate Speech Definitions**

[1] https://www.irishstatutebook.ie/eli/1989/act/19/section/1/enacted/en/html#sec1

[2] https://ec.europa.eu/newsroom/just/document.cfm?doc_id=42985

[3] https://m.facebook.com/communitystandards/objectionable_content

[4] https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy

Social media has become an integral part of daily life for billions of humans across the globe, with social media usage forecasted to grow to 3.43 billion users by 2023 (Dwivedi et al., 2021). Social media platforms such as Instagram, Facebook, Twitter, YouTube, and TikTok allow users to connect, interact, and share content virtually. Research has shown that social media plays a significant role in consuming news (Müller & Schwarz, 2020), adolescent development (Uhls et al., 2017), facilitating the building of relationships and connectivity among humans with mental health conditions (Naslund et al., 2020), and helping to broaden the reach and strengthen the connectivity of contemporary movements such as Black Lives Matter (Mundt et al., 2018). Unfortunately, social media has also provided a platform to disseminate and amplify discriminative material, allowing hateful content to thrive due to policy limitations and the anonymity that these platforms afford (Matamoros-Fernández & Farkas, 2021).

While research and news organizations claim that hate speech in social media is on the rise, little empirical evidence exists in academic research to quantify the prevalence of hate speech in social media. Studies of Twitter have shown that hate speech only contributes towards a minor fraction of the overall tweets. Yet, these platforms' reach and visibility can increase their exposure (Siegel, 2020). This is reflected in a cross-national study in which 53% of Americans, 38% of British, and 31% of Germans surveyed had reported being exposed to hateful content (Hawdon et al., 2017).

This exposure can profoundly impact the health and well-being of targeted victims. Victims are fearful that online threats may translate into real-life violence; these fears are not unfounded, with recent events such as the Charlie Hedbo attack in Paris triggering online and subsequently real-life hate crimes against the Muslim community (Awan and Zempi, 2015). Further studies have corroborated these fears, highlighting exposure to hateful content as a motivation for violence towards targeted groups and correlating with increases in anti-refugee incidents in Germany ( Müller & Schwarz, 2020).

Policymakers and social media companies have responded to these issues by introducing legislation, policies, and procedures to deter users from disseminating hateful content (Pater et al., 2016), with regions such as the United Kingdom making it a criminal offense.  These steps alone are not sufficient to prevent hateful content,  as shown by the exposure levels reported by Hawdon et al. (2017).   The sheer volume of content produced and reported makes it infeasible to solve by a human editorial process; therefore, artificial intelligence is required to detect and classify hate speech (Ullmann & Tomalin, 2020).

## 2.2 Traditional Methods of Hate Speech Detection

The public discourse around hate speech has increased interest within the research community, with a search for "hate speech" on the ACM Digital Library returning over 100,000 documents between 2001 to 2021.

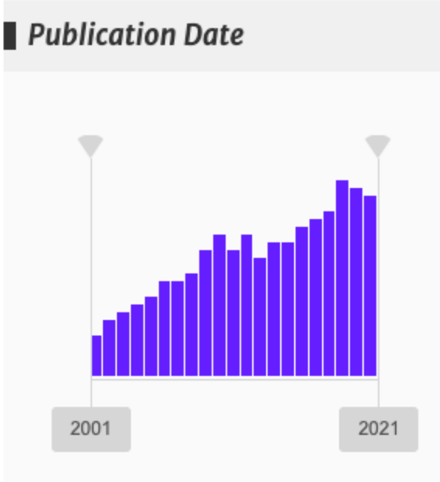


**Figure 2.1:  Distribution of Search Results for Hate Speech on ACM[5]**

The majority of earlier research in this area has treated hate speech detection as a traditional text classification problem.   Solutions generally consist of labelling a dataset of posts from social media with a numeric class label representing a category such as hateful or offensive content.  The labelled dataset is pre-processed to remove or correct problematic content before applying NLP techniques to extract salient features from the text in a numeric format, which can then be fed as input into a supervised machine learning algorithm for training and validation.   This section will

[5] https://dl.acm.org/action/doSearch?fillQuickSearch=false&target=advanced&expand=dl&AfterYear=2001&BeforeYear=2021&AllField=hate+speech

focus on the data curation, pre-processing, feature extraction, and the supervised algorithms employed in the research of hate speech detection and similar text classification domains, along with some of the limitations with these approaches.

### 2.2.1 Hate Speech Data

Twitter is a popular source of data due to its public content and the accessibility of its APIs. The percentage of hate speech in Twitter posts is low, so strategies have been used to increase the occurrence of hate speech in search results, including using known hate terms, divisive topics, or events involving minority groups (Burnap & Williams, 2016; Chen et al., 2012; Davidson et al., 2017). This strategy can lead to racial and dialect bias in datasets (Davidson et al., 2019; Sap et al., 2019). Datasets have also been derived from other social media platforms like Gab (Mathew et al., 2021), YouTube, and Reddit (Mollas et al., 2022 ).

Datasets must be human-annotated, which can be costly, resulting in smaller data samples being used. Research can be outsourced to process larger datasets; however, annotation accuracy can be lower than using specifically trained annotators (Nobata et al., 2016) and raises ethical issues around exposing annotators to toxic data (Sap et al., 2020). Comparing results across research has been challenging due to the lack of an established benchmark dataset (Schmidt & Wiegand, 2017), with the datasets released by Waseem and Hovy (2016) and Davidson et al. (2017) having the greatest adoption (Poletto et al., 2020).

### 2.2.2 Pre-processing

Standard text pre-processing techniques such as converting text to lowercase, splitting sentences into individual word tokens, and lemmatization or stemming are generally applied to data before processing (Power et al., 2018, Davidson et al., 2017, Mathew et al., 2021). Lemmatization and stemming can help reduce the problem dimension space by decreasing the vocabulary size by converting words to a common derivative, e.g., running → run or drank → drink.

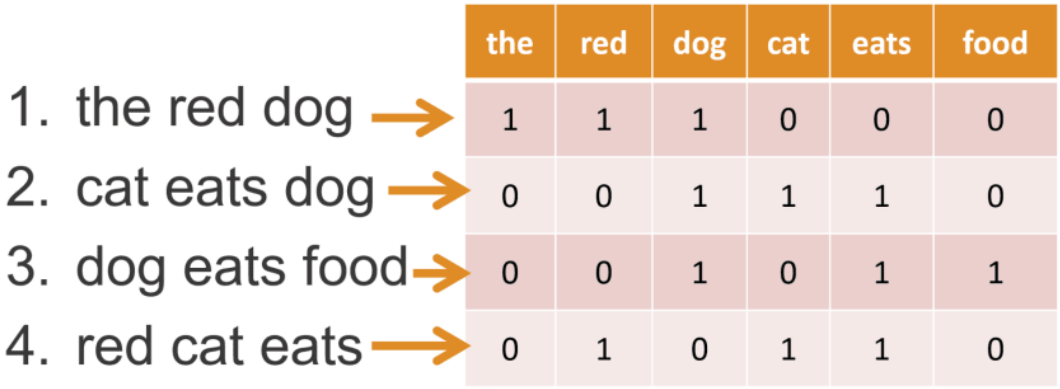
Techniques such as removing non-words like URLs, numbers and common low information words such as “it”, “the” and “so”, referred to as stop words, can further reduce the vocabulary size without impacting sentiment ( Zhao & Gui, 2017). However, valuable context can be removed when these are applied. Social media content can be rife with internet slang, spelling mistakes, and word concatenation via hashtags, often as a deliberate tactic to avoid censorship. Examples include terms such *h8* (hate), *id1ot* (idiot), *fck* (fuck), or #alllivesmater. Techniques to address this include using slang dictionaries to replace known abbreviations and spelling errors (Power et al., 2018) and splitting concatenated phrases (Mathew et al., 2021).

### 2.2.3 Feature Extraction

Surveys on hate speech literature have documented the common use of lexical-based approaches in this field (Schmidt & Wiegand, 2017). This approach works under the assumption that the presence of hateful terms is a strong indicator of hate speech. Text is compared to a dictionary of hateful terms, derived from public repositories such as HateBase[6], with the count of hateful words present in the text being used as a feature. Using this approach has limitations as it fails to capture the nuances of hateful speech and the context of the use of language. This method would flag the term “n*gga” as hateful due to its use as a racial slur; however, it is commonly used as an expression of group solidarity within the African American community (Warner & Hirschberg, 2012). Using this method independently can impact freedom of speech and lead to racial bias, but it can be effective as part of a broader feature set (Chen et al., 2012).

Bag of Words (BoW) techniques have long been used in research. This approach builds a dictionary of words from the dataset and converts sentences into sparse vectors which contain a frequency count of the occurrences of words from the dictionary as features (Djuric et al., 2015; Greevy & Smeaton, 2004). This creates a more dynamic vocabulary but removes context by stripping away the semantics and syntactic structure of the text, resulting in similar misclassifications of hate speech as the lexical approach (Burnap and Williams, 2015).

[6] https://hatebase.org/



| | the | red | dog | cat | eats | food |
|---|---|---|---|---|---|---|
| 1. the red dog → | 1 | 1 | 1 | 0 | 0 | 0 |
| 2. cat eats dog → | 0 | 0 | 1 | 1 | 1 | 0 |
| 3. dog eats food → | 0 | 0 | 1 | 0 | 1 | 1 |
| 4. red cat eats → | 0 | 1 | 0 | 1 | 1 | 0 |

**Figure 2.2: Bag of Words Vectors**[7]

N-grams expand the unigram BoW approach by generating n-length sequences of words, allowing additional context to be captured in the corpus. An earlier survey has shown N-grams to be effective features, with larger n-grams resulting in better classification performance but more computationally expensive (Schmidt & Wiegand, 2017).

Character N-grams, which build N-grams of the individual character combinations within the dataset, have been used to address deliberate misspellings to avoid hate speech detection (Gröndahl et al., 2018; Mehdad & Tetreault, 2016). Interestingly, research has shown that using the presence of these deliberate misspellings as a feature can help improve classification (Nobata et al., 2016).

#### 2.2.3.1 Word Embeddings

Word embeddings address some of the limitations of BoW and N-grams by creating dense low dimension representations of words within the text. These representations capture neighbouring words and context, enabling semantic relationships between words to be learned (Mikolov et al., 2013). Word embeddings such as GLoVE, word2vec, and paragraph2vec have been pre-trained on a large corpus of text and provide the benefit of capturing a richer understanding than dictionaries or embeddings

[7] Sourced from https://www.ronaldjamesgroup.com/blog/grab-your-wine-its-time-to-demystify-ml-and-nlp

built from smaller datasets. Word embeddings also better handle the problem of unseen words which can occur when deploying models on smaller datasets; however, the models don't fully solve for these out of vocabulary words. These pre-trained models have shown gains over traditional BoW models when applied to hate speech detection (Badjatiya et al., 2017; Djuric et al., 2015).

#### 2.2.3.2 Other Techniques

Other novel techniques have been applied for feature extraction. The sentiment of text is often used as a feature to improve classification due to the negative polarity of hate speech (Cao et al., 2020; Chatzakou et al., 2017); however, the presence of positive words within hate speech can be sufficient to circumvent detection via this approach (Gröndahl et al., 2018). "Othering" language has been used as a feature for hate speech classification, with typed dependencies used to extract relationships between words which represent an us versus them dichotomy that is common in hate speech (Burnap & Williams 2015; Burnap & Williams 2016), providing accuracy gains over a standard BoW approach.

### 2.2.4 Classification Algorithms

Earlier research relied upon supervised learning algorithms to train on the features extracted from datasets. Support Vector Machines (Greevy & Smeaton, 2004; Burnap & Williams, 2015), Logistic Regression (Davidson et al., 2017; Waseem & Hovy, 2016), and Random Forest Trees (Burnap & Williams, 2016) have been commonly used for hate speech classification. Support Vector Machines (SVM) and Random Forest Trees were identified as the most frequently used algorithms for hate speech classification in a recent survey (Fortuna & Nunes, 2018), with SVMs generally offering the best performance.

## 2.3 Deep Learning

Convolutional Neural Networks (CNN) are deep learning models which have been commonly applied to image classification problems due to their ability to extract local features from neighbouring pixels automatically and learn patterns (Krizhevsky et al., 2012). Text classification problems have similar challenges in identifying local dependencies and relationships between words across text, leading to experiments with

CNNs for NLP tasks, leveraging word encodings or word embeddings as inputs (Kim, 2014).

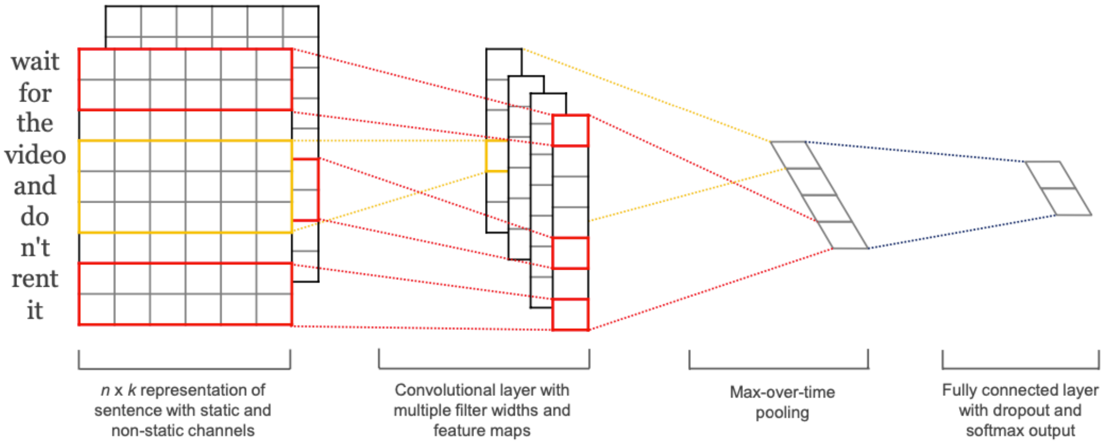

wait
for
the
video
and
do
n't
rent
it
*n* x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

**Figure 2.3: CNN model for sentence classification (Kim, 2014)**

Research has shown that CNNs using word embeddings offer significant performance improvements over traditional BoW and SVMs in detecting toxicity in online comments (Georgakopoulos et al., 2018). Further experiments have applied CNNs to the specific problem of hate speech detection in social media, corroborating improved performance through using word embeddings instead of n-grams (Gambäck & Sikdar, 2017) and highlighting performance gains and variance reduction through ensemble models (Zimmerman et al., 2018).

Recurrent Neural Networks (RNN) offer a distinct advantage over CNNs when modelling text, as they are designed to process temporal sequential information such as sentences. As a result, they are better suited to the NLP problem of classifying and predicting texts. The use of RNNs has been employed to improve hate speech detection. Research has shown that RNNs trained on word embeddings outperform CNNs in detecting abusive online comments for content moderation (Pavlopoulos et al., 2017).

## 2.4 Transformers

The field of computer vision achieved major breakthroughs through the use of transfer learning from large pre-trained models, showing impressive results through fine-tuning models pre-trained with ImageNet datasets (Deng et al., 2009). Researchers in the

NLP domain began exploring opportunities to incorporate inductive transfer learning techniques to NLP tasks beyond the use of word embeddings as features in deep learning networks. Research highlighted impressive results by conducting unsupervised pre-training to generate generalized LSTM models from a large corpus of text which can then be fine-tuned on datasets for specific supervised NLP tasks, with both ULMFit and ELMo outperforming models built specifically for these tasks (Howard & Ruder, 2018; Peters et al., 2018).

The introduction of transformer-based architectures increased the focus on leveraging pre-trained models for transfer learning and resulted in significant advances in the NLP domain. This architecture introduced attention mechanisms that offered faster training times due to parallelization and the ability to learn longer range dependencies in sequences than LSTMs, outperforming the previous state-of-the-art RNNs in sequence and language modelling. Importantly, this architecture also demonstrated the ability to generalize well on other NLP tasks with limited training data (Vaswani et al., 2017).

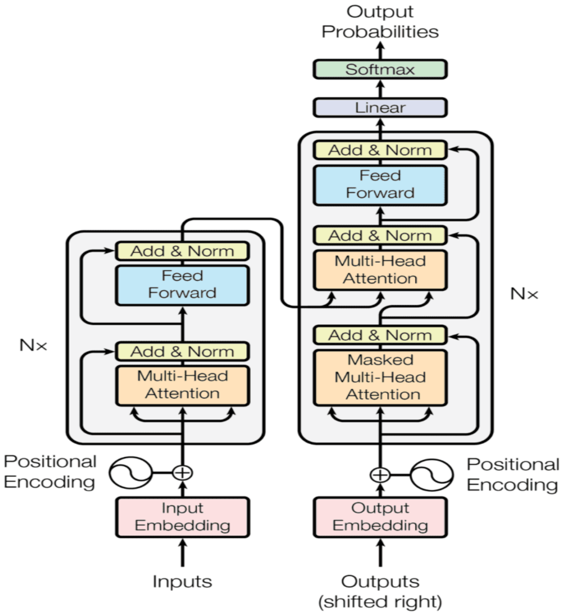


**Table 2.2: Transformer Architecture Diagram**

Subsequent research by OpenAI leveraged the transformer architecture to develop the GPT (generative pre-training) model and built upon the previous ULMFiT and ELMo models by generating pre-trained models followed by supervised fine-tuning for specific NLP tasks, providing a more task agnostic architecture (Radford & Narasimhan, 2018).

### 2.4.1 BERT

The release of the open-source BERT (Bi-directional Encoder Representations of BERT) model by Google marked a significant milestone in the NLP domain. The BERT model leverages a Transformer encoder network to create deep bidirectional representations of terms, their position, context, and semantics for multiple NLP tasks. This model advanced upon the limitations of the unidirectional, left to right architecture of the GPT model by using a masked language model (MLM) to enable bidirectional representations of inputs, increasing the contextual awareness of language models (Devlin et al., 2019). The standard BERT base model consists of 12 encoder layers of transformer blocks, with each encoder containing a multi-headed self-attention layer, with a hidden layer size of 728 and 110m trainable parameters. The larger BERT model has 24 encoder layers, 1024 hidden layers, and 340m trainable parameters. The model is trained on over 16GB of data from English Wikipedia and the Books Corpus (Zhu et al., 2015). The same architecture is used across multiple NLP tasks such as question and answering or text classification with minimal changes to the output layer. This architecture enabled BERT to achieve state of the art results across various NLP benchmark tasks such as SQuAD (Rajpurkar et al., 2016) and GLUE (Wang et al.,2018).

Research in the NLP domain has shifted towards developing improved versions of the BERT architecture. The private technology sector, including social media companies, has made leading contributions. Given the high computational resources, costs, and energy consumption associated with pre-training these large transformer models, this is unsurprising (Strubell et al., 2019). Facebook’s RoBERTa (Y. Liu et al., 2019) and Microsoft’s DeBERTa (He et al., 2021) are variations of BERT which have superseded the original BERT model as state of the art.

### 2.4.2 Application of Transformers in Hate Speech

The public availability of pre-trained state-of-the-art transformer models, combined with their ability to fine-tune on text classifications tasks with relatively small datasets, has seen these increasingly applied to the problem of hate speech detection. SemEval is a series of international NLP workshops that often contain tasks related to detecting hateful comments. The SemEval 2019 workshop consisted of a task to classify offensive language in a social media dataset (OLID). Less than 8% of the 104 submissions leveraged the nascent BERT model, yet these dominated results. Seven out of top 10 models used BERT, including the best performing model (Zampieri et al., 2019).

The following year, SemEval2020 consisted of a similar task but expanded the problem to include multiple languages. Most submissions now used transformer-based architectures, including BERT and RoBERTa. Transformer-based architectures outperformed conventional models, with an ALBERT ensemble producing the best results (Zampieri et al., 2020). SemEval 2021 consisted of a task to detect toxic spans within online comments. Transformer-based architectures continued to dominate submissions and performance, highlighting their position as state of the art in text classification (Pavlopoulos et al., 2021).

Research has continued to experiment with these pre-trained models by optimizing or supplementing transformer architectures through adding LSTM layers or ensemble models with promising results (Pavlopoulos et al., 2021). Mozafari et al. (2020) achieved state-of-the-art precision of 92% using a BERT model fine-tuned using Convolutional Neural Networks (CNN) against the Davidson et al. (2017) dataset.

## 2.5 Transformer Concerns

Research has improved upon BERT-based models' performance by developing increasingly larger and more powerful models trained on a larger corpus of text. The GPT-3 model (Brown et al., 2020) contains a whopping 175 billion parameters and is trained on roughly 45TB of data, an exponential increase on the 340M parameters, and 16GB of data used for the BERT large model. The increasing size of these models raises several concerns. Research by Strubell et al. (2019) has highlighted the environmental cost of these larger models due to the high $CO_2$ emissions from the computational power needed to train them. The authors and Schwartz et al. (2020)

have called for greater investment into more computationally efficient algorithms for a greener AI environment.

The financial cost of training these larger models can also be prohibitive to their adoption (Sahn et al., 2019). For example, the NAS model is estimated to cost somewhere between $940,000 and $3.2 million US dollars to train (Strubell et al., 2019). These increasing model sizes also impact performance speed (Wang et al., 2020), while the financial barriers to entry can limit who can contribute to this field of research, potentially leading to inherent bias within the models (Bender et al., 2021).

## 2.6 Efficient Transformers

A number of lighter, more computationally efficient models have emerged to address concerns on the speed, cost, accessibility, and environmental impact of larger models. Research has produced models such as DistilBERT (Sahn et al., 2019) and MobileBERT (Sun et al., 2020) that provide more compact representations of BERT. This is achieved through a compression technique called knowledge distillation. This process involves training a smaller student model on a larger teacher model to reproduce its behaviour. Student models are trained using the probability distribution of the teachers output, forming “soft” targets, with a cross entropy loss applied to these soft targets instead of the gold labels or “hard” targets. Trained models tend to produce a high probability for the predicted class and a near zero probability for other classes. To address this, a softmax temperature is applied when training the student and teacher to smooth the probability distribution. This process reveals more signals, enabling the student to learn information regarding similarities between classes, often referred to as dark knowlege. These techniques enable the student to generalize in the same fashion as the teacher model.

The DistilBERT model compresses the BERT architecture by reducing the number of layers by a factor of 2 and removing the token-type embeddings and pooler layer. The authors of the DistilBERT paper reported that models are 40% smaller and 60% faster than BERT while retaining 97% of its performance. DistilBERT models have been applied to the classification of hate speech in Twitter data at FIRE 2020, outperforming RoBERTa models on English language tasks (Kumar et al., 2020).

Researchers from Facebook have proposed the Linformer model (Wang et al., 2020), which uses approximations of the self-attention mechanism using a low-rank matrix to optimize speed and memory, achieving improvements of up to 20 fold in model inference. These lighter weight models can offer comparable performance with the BERT base models, but their optimization for speed and size generally comes with a performance penalty compared to the larger models. Further optimization of these models is required to remain viable competition towards the adoption of larger models.

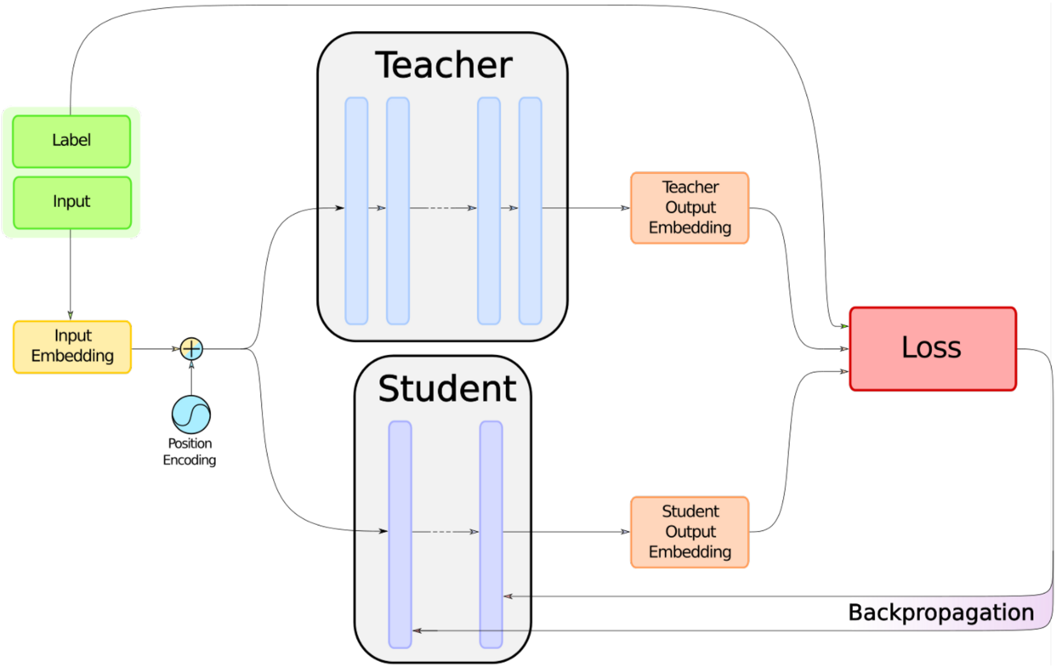


**Figure 2.4: DistilBERT Knowledge Distillation**

## 2.7 Pre-trained Model Fine-Tuning Strategies

Several strategies have been explored to improve the performance of models through applying optimization practices during the fine-tuning of pre-trained models. BERT-based models commonly initialize all layers except a specialized output layer with the pre-trained weights. Lower layers of these pre-trained models contain more general features, while the higher layers are more specialized towards the pretraining tasks. Research by Zhang et al. (2020) highlighted that transferring the weights of higher layers can impact learning and performance when fine-tuning tasks. The research demonstrated improvements in performance when re-initializing between 1 and 3 layers for the BERT base model, with performance plateauing and eventually deteriorating as more layers are re-initialized.

The layers within a transformer architecture each represent different types of information and should therefore have different fine-tuning strategies. Howard et al. (2018) proposed a technique called **discriminative fine-tuning** that uses different learning rates per layer, with the top-most layer having the highest learning rate and the learning rate decreasing through the lower layers. The purpose is to enable lower layers to capture more general features while higher layers are encoded with more localized information to the specific fine-tuning task. Similar techniques are used for the fine-tuning of recent pre-trained models, including XLNet (Yang et al., 2019) and ELECTRA (Clark et al., 2020), and has been referred to as Layer-Wise Decay or Layer-Wise Learning Rate Decay (LLRD).

Recent research has demonstrated the performance benefits of conducting a secondary pre-training stage with a data-rich intermediate task before fine-tuning on the target task (Clark et al., 2019, Sap et al., 2019). Pruksachatkun et al. (2020) performed comprehensive studies to determine which tasks make a good intermediate task across various target tasks. The research concluded that intermediate tasks requiring a high level of inference and reasoning offered the best performance.

## 2.8 Gaps in Research

The conflation of hate speech with offensive language remains one of the prevalent issues throughout research within the NLP domain of hate speech. Governmental and social media hate speech policies aim to address attacks targeted towards minority groups. This is distinguishable from offensive language, which is both unpleasant and generally unwanted, but does not violate hate speech polices.

Datasets commonly used within research reflect this nuance, with annotated classes distinguishing between hate speech, offensive language, and neither. Yet research treats this as a standard multilabel classification task, focusing on improving overall performance rather than identifying specific methods to improve hate speech classification. Previous research has even combined hate and offensive language into a single category, treating the issue as a standard binary classification that offers little academic value.

The absence of a gold standard dataset makes it challenging to compare the results of research. Waseem and Hovy (2016) and Davidson et al. (2017) datasets are the most commonly used yet represent challenges through their method of curation, annotation bias, and imbalanced datasets.  The small percentage of hate speech present in these datasets can limit learning algorithms' ability to identify patterns. Research has argued that this reflects hate speech representing a minority of social media content and, therefore, a justification to neglect techniques such as oversampling or class weight adjustment to address these imbalances.

Pre-trained transformer models such as BERT and RoBERTa offer improved performance with smaller datasets making it an attractive option for the available hate speech datasets.  BERT has been superseded in performance by newer transformer models such as DeBERTA and ERNIE on several NLP benchmarks, which offer scope for further research.   These models are computationally expensive, costly, environmentally damaging, and time-consuming to train,  which can limit hyperparameter searches to find the optimal configuration.  Recent research has used the default configuration of the smaller base models, opting to extend the architecture over comprehensive fine-tuning.

Newer lightweight transformer models, such as DistilBERT, offer comparable performance with BERT, faster training times, and smaller model footprints.  Research into their application towards hate speech classification is limited, and little effort has been applied to evaluating strategies that maximise their performance,  warranting further investigation.

## 2.9 Summary

The literature review discussed in this chapter detailed the challenges in defining and recognizing hate speech due to its complex and subjective nature.  The growth and reach of its occurrence in social media were discussed, along with the negative impact it can have on targeted victims.  This chapter detailed the strategies employed by the academic community to help combat hate speech through automated hate speech detection, with solutions gravitating towards state-of-the-art transformer models such as BERT.  The numerous concerns and challenges with the adoption of ever-growing models were discussed, highlighting the research community's calls to explore more computationally efficient alternatives.  This chapter documented recent attempts by the research community to respond to this call by developing more compact models such as DistilBERT and the opportunities to optimize further, concluding with an overview of the gaps in the research.

# 3 DESIGN AND METHODOLOGY

This chapter provides an overview of the project approach, followed by the design aspects, methodologies, and technologies used during the implementation of experiments performed within this study. This section will also provide a comprehensive description of the datasets used in this research and an exploratory analysis of their data. The section will conclude with an overview of the experiments conducted, their design, and how their performance will be evaluated.

## 3.1 Project Approach

This research aims to evaluate the impact of fine-tuning strategies towards improving the performance of a DistilBERT transformer architecture in classifying hate speech in social media posts from Twitter and Gab. The goal is to determine if a DistilBERT model can outperform a BERT base model and make compelling arguments for lighter weight, faster and cheaper alternatives to the more costly large transformers in the domain of hate speech classification.

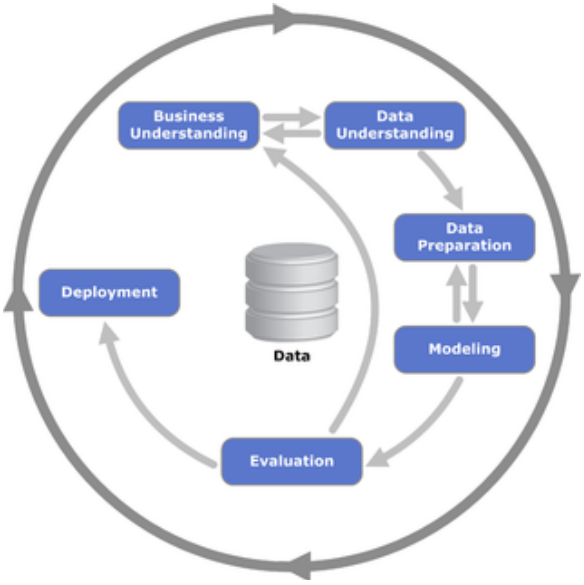


**Figure 3.1: Phases of the CRISP-DM framework**

### 3.1.1 Project Methodology

The project implementation can be divided into five main stages, which align with the CRISP-DM framework. The first stage reflects the *Business Understanding* phase, covered by the literature review in Chapter 2. The second stage aligns with the *Data Understanding* phase, which involves a review and exploratory analysis of the chosen datasets. The third stage, detailed in sections 3.4 and 3.5, addresses the *Data Preparation* phase by conducting the necessary data scrubbing, merging, and pre-processing before fine-tuning the transformer models.

The fourth stage will consist of the *Data Modelling,* which requires fine-tuning the pre-trained BERT and DistilBERT vanilla configurations for the sequence classification of the pre-processed dataset to serve as benchmarks for performance. Further DistilBERT models are generated to assess the individual, and collective performance impact of the weight re-initialization, LLRD, and intermediate task transfer fine-tuning techniques.

The fifth and final stage, the *Evaluation* phase, consists of analyzing and comparing the performance differences in hate speech classification of each generated model. The performance will primarily be measured using the macro f1 score of the multi-class classification, and differences will be tested for significance using statistical tests. Accuracy, precision, and recall will be measured to provide further insights into performance. The results will be reviewed to address the research questions and objectives in sections 1.2 and 1.3, summarized below.

- Is there a difference in the performance of the multiclass classification of hate speech in tweets between a fine-tuned BERT and DistilBERT model?
- Does re-initializing the higher layers of the DistilBERT model improve the performance?
- Does applying higher learning rates to the topmost layers when fine-tuning a DistilBERT model improve performance in the classification of hate speech?
- Does fine-tuning a DistilBERT model on an intermediary task using a larger dataset before fine-tuning on the research dataset improve the performance of the multiclass classification of hate speech?
- Which model performs the best in terms of accuracy, precision, recall, and f1-score for classification of hate speech in the multiclass research dataset?

## 3.2 Design Aspects

### 3.2.1 Model Selection

The initial goal of this research had included the evaluation of the Linformer model instead of the DistilBERT model due to its use on the Facebook and Instagram platforms to detect hate speech. The lack of availability of pre-trained weights rendered the use of Linformer cost-prohibitive. The model was trained on a similar corpus and infrastructure to the BERT base model, which is estimated to cost up to $12,571 to train (Strubell et al., 2019). DistilBERT was chosen as an alternative due to its availability of pre-trained weights, previous use within the research domain, and community support.

| **Model** | **Hardware** | **Training Hours** | **Cloud Compute Cost** |
|---|---|---|---|
| Transformer$_{base}$ | P100x8 | 12 | $41-$140 |
| Transformer$_{big}$ | P100x8 | 84 | $289-$981 |
| BERT$_{base}$ | **V100x64** | 79 | **$3751-$12,571** |
| BERT$_{base}$ | TPUv2x16 | 96 | $2074-$6912 |
| Linformer | **V100x64** | - | - |

**Table 3.1: Model Training Costs (Strubell et al., 2019)**

### 3.2.2 Software and Environment Used

The selection of the DistilBERT model influenced the decision to use the Hugging Face Transformer libraries (Wolf et al., 2020), as the model emanated from the Hugging Face research team. These libraries provide unified APIs to generate task-specific transformer architectures for sequence classification using pre-trained weights for DistilBERT and BERT. The libraries also provide tokenization APIs for data preparation along with trainer interfaces to greatly simplify the development of hyperparameter searches, model training, validation, and evaluation. Comprehensive documentation and support are available via their website[8]. This research used version 16.4.2 of the Pytorch libraries for the experiments conducted.

[8] https://huggingface.co/docs/transformers/index

The analysis, data preparation, experiments, and evaluations were conducted using Python in Jupyter notebooks. Links to the codebase for each experiment can be found in Appendix A of this document. The Google Colabs environment was chosen over SageMaker to run experiments due to ease of use and free computational resources, including access to GPU devices. A Colab Pro+ subscription was ultimately necessary to avail of longer runtimes, background execution of notebooks, and concurrent access to faster GPUs for $49.99 per month. The experiments in this study were thus performed on Tesla P100 GPUs with 53GB of RAM.

### 3.2.3 Data Selection

This research combined labelled datasets from previous research (Mathew et al., 2021; Davidson et al., 2017) to generate a larger dataset for training and improve the model generalization by including data from multiple social media platforms. A large Twitter dataset produced from research by Founta et al. (2018) was considered for inclusion; however, the research author had stipulated that data could be used but not shared. This would impact the ability to reproduce this research and use the aggregated dataset for future research; therefore, this dataset was excluded from this research.

## 3.3 Data Understanding

This research project combines two datasets derived from previous research on hate speech to fine-tune and evaluate the model. The first dataset selected was generated by Davidson et al. (2017). This dataset comprises 24,802 tweets that have been labelled as either hate speech, offensive, or neither. The tweets were curated using search terms derived from a lexicon of hate speech terminology compiled by Hatebase.org. A subset of 25K tweets was randomly selected from a search result set of 85.4 million tweets. Tweets were manually labelled using the annotation crowdsourcing platform CrowdFlower. Workers were instructed to label tweets as either hate speech, offensive but not hate speech, or neither hate speech nor offensive. Workers were given guidance that offensive words may not constitute hate speech depending on the context, thus avoiding the conflation of hate speech and offensive language. Each tweet was labelled by three or more workers, with the majority decision used to mark the tweet. The majority of tweets in the dataset were labelled as offensive, with only

5% labelled as hate speech. This illustrates the limitations of using a purely lexical-based approach for hate speech and it results in a highly imbalanced dataset.

The secondary dataset is the HateXplain dataset proposed as a benchmark by Mathew et al. (2021). The dataset comprises 19,229 records containing a combination of posts from the Twitter and Gab social networks. A set of search terms was generated by combining the lexicons provided by Davidson et al. (2017), Ousidhoum et al. (2019), and Mathew et al. (2019). The resulting tweets from the search responses were randomly filtered from the period Jan-2019 to Jun 2020. The dataset provided by Mathew et al. (2019) was used for the Gab posts. Posts containing links, pictures, or videos were excluded to ensure the context of the post was encapsulated entirely within the text.

The dataset was labelled using Amazon Mechanical Turk to crowdsource annotators. Three annotators labelled posts as either hateful, offensive, or normal, and the majority vote was used to determine the label. 919 posts had an undecided majority label and were excluded from the dataset. The dataset results in a more balanced set, with hateful content representing 30% of posts, albeit predominantly within the Gab content, as hateful tweets only represent 3% of the Twitter data. The dataset also includes information regarding the target groups of hateful content and the rationale of annotators, which were excluded from the experiments in this project.

## 3.4 Data Exploration

### 3.4.1 Twitter Dataset

The Twitter dataset comprises 24,783 records. The majority of records have been classified as offensive language, representing 77.4% of the dataset. Tweets classified as neither offensive nor hateful represent the second largest group, containing 16.8% of labels. Hate speech remains the minority label, with only 5.8% of records classified as hateful.

| **Class** | **Count** | **Unanimous Votes** | **Highest Frequency Terms** |
|---|---|---|---|
| hate_speech | 1430 | 263 *(17.7%)* | faggot, bitch, nigga, nigger |
| offensive_language | 19190 | 14347 *(74.7%)* | bitch, hoe, bitches, nigga |
| neither | 4163 | 2872 *(68.9%)* | trash, bird, yankee, yellow |

**Table 3.2: Twitter Hate High Frequency Terms**

The subjective nature of hate speech classification is highlighted by the different views of annotators, with only 18% of hate speech records being unanimously classified by all annotators. Comparing this with the 75% and 69% of unanimous votes received for offensive and non-offensive posts respectively, demonstrates that hate speech can be more open to interpretation even when equipped with the clear guidelines provided to annotators.

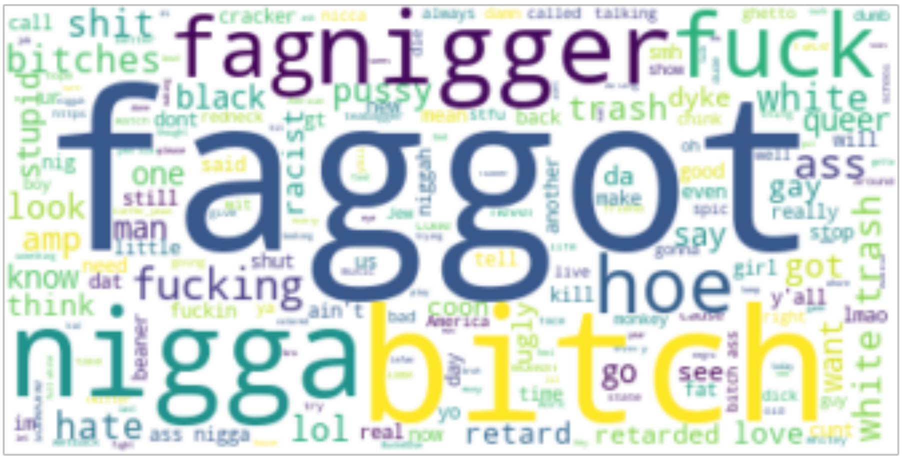


**Figure 3.2: Word Cloud of Twitter Hate Speech Terms**

An analysis of the word frequency within the hate speech records highlights the groups frequently targeted, with homosexual, misogynistic, and racial slurs the most prevalent. Posts classified as offensive contained similar hateful terms, with a higher frequency of misogynistic terms. The context of their use was deemed to be merely offensive as the terms may not have been targeted at a particular group, highlighting the nuance and subjectivity involved.

| **Class** | **Sentence** |
|---|---|
| **Hate** | *“<user> dont tell me what to do. fuck balls kike nigger cunt tits cocksucker chink spic piss bitch bastard pussy faggot.”* |
| **Hate** | *“<user> Dumb Haitian fake black faggots. Go to Haiti and neck yourself.”* |
| **Hate** | *“<user>*1) *He's a faggot and I don't like him. 2) I'm on the other side of the state.”* |
| **Offensive** | *“Cruising in my go kart at walmart selling cupcakes, go ahead admit faggot, this shit is tighter than butt rape”* |
| **Offensive** | *“I guess this is the night bitches die...!!!!" Stewie is that nigga...!”* |
| **Normal** | *“momma said no pussy cats inside my doghouse”* |
| **Normal** | *“<user> Peel up peel up bring it back up rewind back where I\'m from they move Shaq from the line" ooooow who tf said that trash!!?”* |

**Table 3.3: Sample sentences from the Twitter dataset**

### 3.4.2 HateXplain Dataset

The HateXplain dataset comprises 19,229 records, with a more balanced distribution of the class labels than the Twitter dataset. The majority of records have been classified as normal, representing 40.63% of the dataset. Tweets classified as hateful are the second largest group, containing 30.86% of records. Posts labelled as offensive represent the remaining 28.50% of the data.

| **Class** | **Count** | **Unanimous Votes** | **Highest Frequency Terms** |
|---|---|---|---|
| hate speech | 5935 (*30.86%)* | 2960 *(49.9%)* | nigger, kike, jew,white, faggot |
| offensive | 5480 (*28.50%*) | 1761 *(29.67%)* | retarded, white, ghetto, bitch |
| normal | 7814 (*40.64%*) | 5124 *(86.33%)* | white, women, muslim, black |

**Table 3.4: HateXplain Hate High Frequency Terms**

Analysis of the annotator subjectivity sees similarly high agreement across posts classified as normal, with annotators agreeing on 86.6% of posts. Hateful content saw a higher agreement among annotators than the Twitter dataset, with annotators agreeing on 49.6% of hateful content. Offensive posts seemed to cause greater subjectivity with annotators only agreeing on 29.67% of posts.

A review of the common terms in hateful content sees similar trends to the Twitter dataset with racial and homophobic slurs frequently occurring, as illustrated by the word cloud in Figure 3.3. This dataset also prominently features anti-Semitic terms such as “kike” and “jew”, along with the higher frequency of the word “white”.

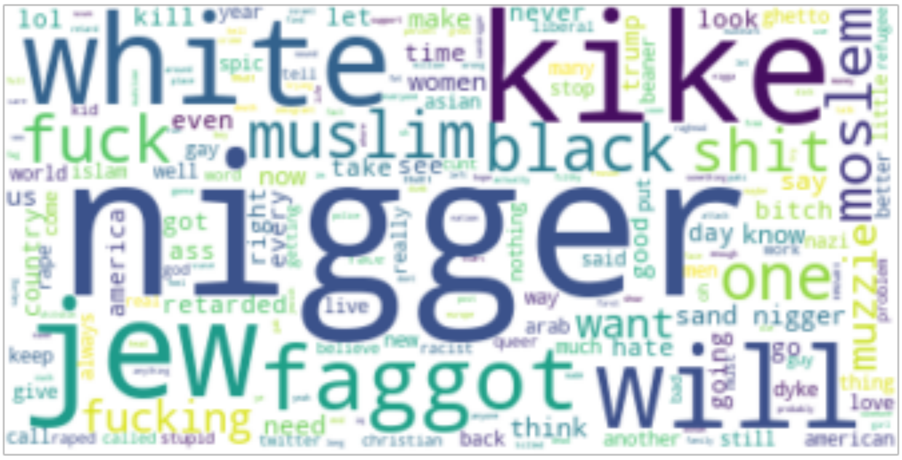


**Figure 3.3: HateXplain Word Cloud of Hate Speech Terms**

The breakdown in Table 3.5 of the Twitter and Gab content contained within the HateXplain dataset highlights that the majority of the hateful and offensive content stems from the Gab dataset.

| **Class** | **Twitter** | **Gab** | **Twitter Terms** | **Gab Terms** |
|---|---|---|---|---|
| hate | 699 | 5227 | nigger, faggot, white | nigger, kike, jew, white |
| offensive | 2323 | 3152 | bitch, faggot, retarded | white, jew, women, ghetto |
| normal | 5756 | 2044 | white, nigga, women | white, women, moslem |

**Table 3.5: Twitter and Gab High Frequency Terms**

| Class | Sentence |
|---|---|
| **Hate** | *"ah k white nationalism is perfect and infallible and you are a cuck faggot who takes it up the ass from blacks if you disagree in any way that seems to cover it"* |
| **Hate** | *"a nigger bich will never have strait hair so keep wearing dat horsehair weave"* |
| **Hate** | *"hell yeah proof that the anti q trolls here are fake alt right kike dick suckers"* |
| **Offensive** | *"birds of prey listening to margot robbie talk like a retarded white girl who thinks she can drop the n word any time she wants for <number> hours hard pass"* |
| **Offensive** | *"can this bitch from saudia arabia be deported already"* |
| **Normal** | *"<user> why should you ban immigrants from operating small business every business start somewhere and tomorrow that will give people job"* |
| **Normal** | *"<user> welcome aboard we are a land of refugees from the p c culture that only p c when they want to be"* |

**Table 3.6: Sample sentence from the HateXplain dataset**

## 3.5 Data Preparation

The HateXplain dataset was downloaded from the Hugging Face Datasets Hub and pre-segmented into an 8:1:1 split ratio for train, test, and validation data. The HateXplain dataset has already been pre-processed by its authors before its publication, with the raw posts unavailable in the datasets. Reposts and duplicates were filtered by the authors from the dataset to maintain data quality.

The authors leveraged the **ekphrasis** library developed by Baziotis et al. (2017) as part of their text processing pipeline in their research on the sentiment analysis of tweets. The library contains text processing tools geared towards social networks, providing tokenization, word normalization, word segmentation, and spelling correction. The library was used to split posts into word tokens and substitute user names, numbers, URLs, and dates with replacement tokens (e.g., <user>, url> ). Contractions were extended into complete form  (e.g., “can’t”  extended into “can not”).  Hashtags containing combined words were separated into individual words, with tags such as “#ingodwetrust” expanded to “in god we trust”.  The authors did not filter emojis as they assumed they would add important information for hate and offensive task classification; however, their contribution towards classification accuracy was not assessed during their research.  The **ekphrasis** library transforms commonly used emojis into tokens.  For example, the laughing emoji (😀) is translated into the token <laugh>.

The Twitter dataset provided tweets in their raw formats, so it requires further pre-processing before model training.  As the data will be combined with the HateXplain dataset, the ekphrasis library is used for consistency.  Usernames, numbers, URLs, and dates are replaced in the text with generic tokens, hashtags and word contractions are extended into their complete form where possible and emojis are replaced with word tokens representing their meaning.  Additional processing is applied to the Twitter data to correct spelling mistakes and replace elongated words with the ekphrasis library using a statistical model derived from the English Wikipedia and 330 million tweets to identify corrections.  Stop words are retained, and techniques such as stemming and lemmatization are avoided to retain the context of the tweets.

The dataset is heavily imbalanced, with hate speech only representing 5.8% of the data set, while the Twitter data is also similarly under-represented in the HateXplain dataset.  To address the imbalance, the Twitter hate speech labelled data is randomly oversampled to increase its count from 1430 to 4000, bringing its percentage to 14.35%.   Random Under Sampling of the majority label, offensive, is avoided to prevent loss of valuable data from a relatively small dataset.  Post-processing, the Twitter and HateXplain datasets are merged to form the combined *HateTwitGab* dataset.

Pre-trained BERT-based models require the textual data to be converted into a specific format.  The Hugging Face Tokenizer classes for the BERT and DistilBERT models were used to convert the sentences into the correct format.  Sentences need to be represented as a single fixed-length vector containing a numerical representation of each word token.  Sentences that exceed the maximum fixed size are truncated to the maximum sentence length. For sentences shorter than the maximum length, vectors are padded out with a special [PAD] token to reach the fixed sentence length.

The BERT-based models assigned a unique ID for each word token encountered during pre-training. Each token of the input data was encoded using the BERT token ids. As these models were trained on a fixed vocabulary from a corpus of text, there is a possibility that applying these encodings will result in words being encountered that were not part of the trained vocabulary. This is commonly referred to as an out-of-the-vocabulary (OOV) problem. These unseen tokens can be converted into a special [UNK] token representing an unknown word. This can strip a significant amount of valuable information from a sentence. BERT addresses this by using the Word Piece algorithm (Wu et al., 2016) to break words into sub-words, allowing common sub-words to be encoded. For example, the word 'snorting' is broken down in to 'snort' and '##ing', with the first token representing a known sub-word and the second token prefixed with 2 hashes to indicate it is suffixed by a sub-word.

Input sequences require two additional special tokens to be included, [CLS] and [SEP]. The CLS token, standing for classification, is added to the beginning of the sentence and is used to encode a representation of the meaning of the entire sentence. The [SEP] token, which stands for separator, is used to mark the end of a sentence. An additional vector is generated for each sentence, called the attention mask, which contains a binary representation of each token. A value of 1 is used to tell the model that this is a real token that needs to be attended to, and a value of 0 for those tokens such as [PAD] tokens that can be ignored.

## 3.6 Modelling

The research aims to evaluate the performance gains achieved in hate speech classification with the DistilBERT model by optimizing the fine-tuning process. The goal is to demonstrate that a cheaper, lighter weight, and faster model can offer a viable alternative to the state-of-the-art BERT model. The first stage will develop benchmarks for performance comparison by fine-tuning both the BERT and DistilBERT models for sequence classification of hate speech using the *HateTwitGab* dataset. A preliminary hyperparameter search will be performed on the DistilBERT model to determine the optimal configuration for fine-tuning.

The second stage will consist of applying several fine-tuning strategies independently to optimize the performance of DistilBERT models, including layer-wise learning rate decay, weight re-initialization, and intermediate task transfers. The third stage will combine strategies that demonstrate performance gains to develop an optimized classification model, with further hyperparameter searches performed to optimize configuration.

### 3.6.1 Baseline Models

To create a comparative baseline for performance, vanilla versions of both BERT models and DistilBERT are fine-tuned on the *HateTwitGab* training data using the Hugging Face Transformer libraries.

#### 3.6.1.1 BERT Baseline

The pre-trained BERT base model is used for fine-tuning on the classification of the *HateTwitGab* data. The uncased version is used, which coverts text to lowercase prior to tokenization. The BERT model consists of 12 encoders with 12 bidirectional attention heads, 768 hidden size, and 110 million parameters. A larger BERT model was used to achieve the state-of-the-art performance reported by Devlin et al. (2019), consisting of 24 encoder layers, 16 attention heads, and 340 million parameters. This model was considered too costly from a performance perspective for this research. For the multi-class sequence classification tasks, a simple linear layer with a softmax activation function is added above the output pool in the BERT base model with an output size of three.

The model is trained using the input ids, attention masks, and target labels of the tokenized Twitter training dataset as inputs. The model is trained over three epochs using the AdamW optimizer with an initial learning rate of 1e-5 and a cross-entropy loss. The model is trained in batch sizes of 32, which controls the amount of data processed before internal parameters are updated during the training process. The model with the lowest validation loss across the three epochs was evaluated for performance against the *HateTwitGab* test dataset.

#### 3.6.1.2 DistilBERT Baseline

The base uncased pretrained DistilBERT model is used to develop a benchmark model to compare against optimized versions of this model architecture. The model consists of 6 encoding layers, 12 attention heads, 768 hidden size, 66 million parameters and a linear classification layer with a softmax activation function for sequence classification. The AdamW optimizer and cross-entropy loss were used, as with the BERT benchmark model.

A preliminary hyperparameter search is performed to determine the optimal settings for parameters, including the learning rate, batch size, random seed, and the number of epochs. The Optuna hyperparameter framework is used instead of a standard grid search, as it helps accelerate the process through efficient search and pruning strategies. The optimal values identified during the hyperparameter search trials are applied when fine-tuning the DistilBERT model to define the baseline for performance.

### 3.6.2 Optimization Strategies

A series of experiments are conducted to evaluate the impact of optimization strategies on the performance of the DistilBERT models for sequence classification. All models initially use the same training configuration as applied to the baseline DistilBERT model.

#### 3.6.2.1 Re-initializing Pretrained Layers

Experiments are conducted to evaluate the performance impact of weight re-initializing on DistilBERT models when fine-tuning the model with between one and six layers re-initialized. The implementation of the DistilBERT model uses a normal distribution of values with a mean of 0 and a standard deviation of 0.02 to initialize weights. An algorithm was developed to iterate through the desired number of layers and apply this initialization logic.

#### 3.6.2.2 Layer-Wise Learning Rate Decay

LLRD is applied to the DistilBERT models to assess the performance impact on hate speech sequence classification. This was achieved by developing an algorithm to set a peak learning rate of 1e-5 to the topmost layer of the DistilBERT model and applying a multiplicative decay factor to decrease the learning rate by each layer from top to bottom. Multiple models are fine-tuned to identify the optimal decay rate from a range of 0.95 to 0.7.

#### 3.6.2.3 Intermediate Task Transfer

To assess the performance impact of fine-tuning a model on an intermediate task before fine-tuning on the hate speech classification task, a DistilBERT model is trained on the SQuAD question and answering task (Rajpurkar et al., 2016). The SQuAD task was chosen due to its large dataset size (98,169), and its requirement for complex reasoning and inference, factors flagged as working best in previous research (Pruksachatkun et al., 2020). The SQuAD dataset is retrieved from the Hugging Face datahub and is pre-processed into question and answer pairs. The DistilBERTForQuestionAnswering model from the Hugging Face library is used, consisting of a DistilBERT model with a span classification head on top. The model is trained for three epochs on the processed SQuAD dataset.

The epoch with the lowest validation loss is saved as the best performing iteration of the model. Its weights are loaded into a new DistilBERT model created for sequence classification. This model is further fine-tuned upon the *HateTwitGab* data for the hate speech classification target task over three epochs.

#### 3.6.2.4 Optimal DistilBERT model

Four DistilBERT models are created to assess the performance implications of combining the various optimizations strategies evaluated independently. The models are fine-tuned on the *HateTwitGab* dataset. The best performing model from the experiments is subjected to a hyperparameter search to evaluate if alternate configurations are better suited to the optimized model.

## 3.7 Performance Evaluation

To evaluate individual models, this study reports on the accuracy, precision, recall, and the macro F1 score. These metrics are commonly used for multi-class classification performance evaluation and have been used to evaluate the performance of hate speech classification by Matthew et al. (2021), Davidson et al. (2017), and Mozafari et al. (2020). The macro F1 score is preferred as the primary metric for evaluation, given the imbalanced nature of the dataset and the desire to place equal importance on the classification of the hate speech label. These metrics will be rounded up to two decimal places when presented in this research. The following section provides further information on what these metrics capture and how they are calculated.

**Precision:** This captures the ratio between the number of true positives and the total number of predicted positives. This defines how many outcomes were correctly identified as hate speech out of the total number of outcomes predicted as hate speech.

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)}$$

**Recall:** This metric defines a model's ability to predict true positives. Using the hate speech label as an example, this metric calculates the percentage of outcomes correctly predicted as hate speech from the total number of hate speech outcomes.

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)}$$

**Accuracy:** This metric defines the number of correct predictions from the total number of outcomes. For the hate speech label, this would calculate the number of outcomes correctly predicted as hate speech, plus those correctly identified as not hate speech from the total set of predictions.

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{True\ Positive\ (TP) + True\ Negative\ (TN) + False\ Positive\ (FP) + False\ Negative\ (FN)}$$

**F1 Score:** The F1 score represent the harmonic mean of precision and recall, providing a more balanced view of model performance.

$$F1\ Score\ =\ 2 * \ \frac{Precision * Recall}{Precsion+Recall}$$

A micro, weighted or macro F1 score can calculate the model score across the three outcome labels. The micro average essentially calculates the model accuracy and is rarely used as a classification metric. The weighted average calculates the weighted mean of the F1 scores across all labels and can be undesirable for imbalanced datasets, given the influence of majority labels on the scoring. The macro F1 score uses the unweighted mean F1 score across labels, ensuring all outcome labels have equal weight. The macro average was selected for this research due to the imbalanced dataset and the importance of reflecting the model's ability to detect the minority hate speech label. A confusion matrix for each model will be generated to compare and contrast a model's performance across the labels hate_speech, offensive and normal.

For model comparisons, models will be trained and tested against the *HateTwitGab* dataset, performing ten iterations against different random training data splits to eliminate bias. The macro F1 score will be recorded for each iteration, and the distribution of scores will be compared using statistical analysis tools. The model score distributions will be assessed for normality using scatter plots, and the Shapiro Wilk test is used to assess normality at the 0.05 alpha level.

A dependent t-test will assess if the variations between model performance are significant if the distribution is normal; otherwise, the Wilcoxon Signed-Rank test will be used. These tests were selected as the data splits used across each test are not independent due to the random sampling of the same dataset. One-tailed tests will be used in both cases as we are testing for a performance improvement. The null hypothesis will be rejected if the tests indicate significant improvement at a 0.05 alpha level. Otherwise, the null hypothesis will be accepted. The rejection of the null hypothesis provides evidence that a DistilBERT model fine-tuned with optimization strategies can provide superior performance to a vanilla BERT model and should be considered a viable tool for hate speech classification in future research.

# 4 RESULTS, EVALUATION AND DISCUSSION

This chapter provides a detailed overview of the implementation of the six experiments described in section 3.6, including the data preparation required in advance of these being conducted. The results of each experiment are presented, with the performance metrics detailed in section 3.7 used to evaluate the performance of each model. The implementation and the results of the statistical tests used to assess the significance of the findings are discussed. The chapter will conclude with a discussion summarizing the results of the experiments and evaluate the evidence to confirm the null hypothesis.

## 4.1 Data Preparation

To prepare for the model training, the Twitter and HateXplain datasets first needed to be pre-processed to remove or replace unwanted words or characters. The HateXplain dataset was sourced from the Hugging Face Dataset hub and was already split into a train, validation, and test ratio of 8:1:1. As this dataset would need to be merged with the Twitter dataset and splits randomized, the HateXplain data splits were merged into a single dataset.

The HateXplain data had already been pre-processed by the authors using the **ekphrasis** library, which cleans the text and replaces problematic components with generic token tags. The output is converted into a list of tokens, thus requiring no further manipulation of the content. The columns *id* and *rationales* were irrelevant for this research and were dropped from the dataset after the data exploration. The class label was extracted from the *annotators* column. This column provided a list of the labels selected by each annotator; therefore, the mode was used to determine the class label.

| **Twitter Label** | **Twitter Encoding** | **HateXplain Class** | **HateXplain Encoding** | **HateXplain Modified Encoding** |
|---|---|---|---|---|
| hate_speech | 0 | hatespeech | 0 | 0 |
| offensive_language | 1 | offensive | 2 | 1 |
| neither | 2 | normal | 1 | 2 |

**Table 4.1 : Dataset Label Encodings**

The label encodings in HateXplain differed from the encodings used within the Twitter dataset and were therefore converted for consistency, as illustrated in Table 4.1, with the Twitter format preferred.

The tweet column in the Twitter dataset contained the original tweet text, requiring pre-processing prior to use for training. The **ekphrasis** library was used for consistency with the HateXplain dataset, using the modifications detailed in section 3.5.

As the Twitter dataset was imbalanced, with hate_speech being the minority label, the data was randomly over-sampled as described in section 3.5. Both datasets were merged forming a single dataset which will be referred to as *HateTwitGab*, containing the columns *sentence* and *label* and the below class distribution.

| **Total Records** | **hate_speech** | **offensive** | **normal** |
|---|---|---|---|
| 46,582 | 9,935 (21%) | 24,670 (53%) | 11,977 (26%) |

**Table 4.2: HateTwitGab Class Distribution**

The sentence column in the resulting dataset was tokenized using the Hugging Face DistilBertTokenizer, which runs the end-to-end tokenization, including punctuation splitting and the wordpiece sub-word generation. Analysis showed that greater than 99% of the sentences had 64 tokens or less, while the majority were between 10 and 40. The value of 64 was chosen as the maximum length for the sentence encodings. Sentences with more than 64 tokens were truncated, and sentences that fell below 64 tokens were padded to the maximum length.

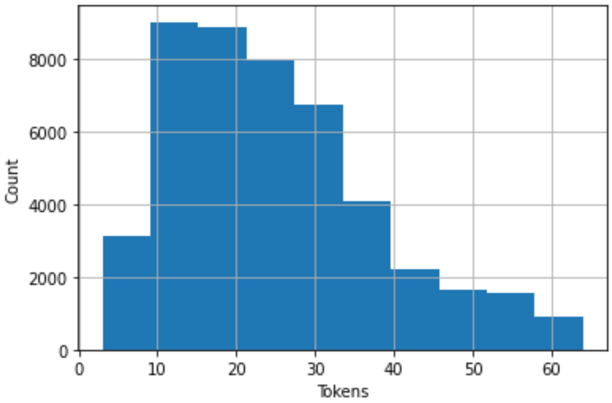


**Figure 4.1: Token count distribution**

The outputs of the tokenization process, the input_ids, and attention_mask were added to the *HateTwitGab* dataset. This tokenization process was repeated with the Hugging Face BertTokenizer as the BERT model contains different embeddings to DistilBERT as it was trained on a larger corpus. The input_ids and attention_mask data columns generated by the BERT tokenization were appended to the *HateTwitGab* dataset, respectively as input_ids_bert and attention_mask_bert.

To reduce the effect of data sequencing influencing performance, each model needed to be trained, validated, and tested against ten randomly sorted datasets. The dataset was randomly shuffled ten times and split into the train, validation, and test ratio of 8:1:1. Splits were stratified across the class labels, given the imbalanced nature of the dataset. The outputs of each of these were persisted as *HateTwit1.csv to HateTwit10.csv,* the first of these will be referred to as *HateTwitGab_1*. These saved splits were reused across each experiment to enable the reproducibility of the research.

## 4.2 Experiment 1: BERT Baseline

To establish the baseline that will be referenced to confirm or reject the null hypothesis, experiments were conducted to evaluate the performance of a vanilla BERT model on the *HateTwitGab* dataset. The **BertForSequenceClassification** model from the Hugging Face Transformer library was used to load the pre-trained **bert-base-uncased** model, the base version of BERT with 12 encoders, 12 bidirectional attention heads, 768 hidden size, and 110 million parameters. The below arguments were used as the training parameters for the experiment, using some of the recommended settings provided by Devlin et al. (2019).

- Training Batch Size = 32
- Evaluation Batch Size = 32
- Epochs = 3
- Learning Rate = 1e-3
- Random Seed = 2

The model was trained and evaluated against the ten variations of the training, validation, and test splits described in section 4.1. Some minor pre-processing was necessary for the datasets before training to correct the renaming of the BERT tokenizer outputs generated in during the data preparation stage. For each iteration, 80% of the data was used to train the model, with 10% to validate the model. The model output from the epoch with the lowest valuation loss was evaluated using the remaining 10% of the dataset, with all 10 iterations recorded. Table 4.3 details the mean and standard deviation of the performance metrics, along with thee values of the best and worst performing models from the test iterations.

| Metric | Macro F1 | Accuracy | Precision | Recall |
|---|---|---|---|---|
| Mean | 74.05 | 78.77 | 74.46 | 74.09 |
| Standard Dev | 0.63 | 0.58 | 0.76 | 0.60 |
| Best Model | 74.89 | 79.63 | 75.78 | 74.82 |
| Worst Model | 73.29 | 78.12 | 73.51 | 73.40 |

**Table 4.3: BERT Classification Average Performance**

The BERT baseline achieves an average macro F1 score of 74.05%, with precision and recall also in the 74% range. There is little volatility in the model training over the 10 iterations, with a low standard deviation of 0.63 across the macro F1 scores.

A review of the model's average performance against each of the class labels gives greater insight into the model's ability to classify hate_speech, and distinguish from offensive language. The model scores well in classifying offensive language with an average F1 score of 86.1%. The model is less successful in the classification of hate speech, scoring 74.40%, with a lower precision of 70.64% indicating that the model may have overfitted on the hate label.

| Label | F1 | Precision | Recall |
|---|---|---|---|
| Hate Speech | 74.40 | 70.64 | 78.61 |
| Offensive | 86.08 | 85.58 | 86.59 |
| Normal | 61.68 | 67.17 | 57.08 |

**Table 4.4: BERT Classification Confusion Matrix**

Analysis of the training and validation loss for the best and worst performing models shows that the validation loss starts to overfit after the first training epoch, indicating a limited learning capacity over a higher number of epochs.

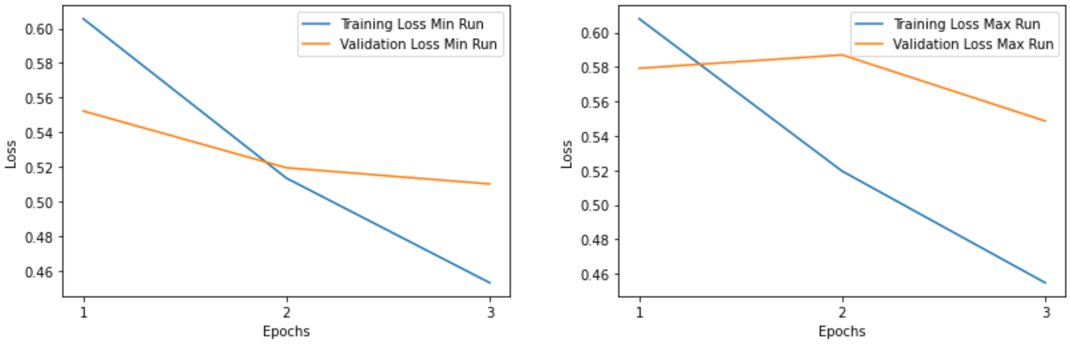


**Figure 4.2: BERT Fine-tuning Validation Loss**

## 4.3 Experiment 2: DistilBERT Baseline

A secondary baseline was required to compare the performance impact of optimizations to the DistilBERT fine-tuning process. A DistilBERT model was created using the **DistilBertForSequenceClassification** model from the Hugging Face Transformer libraries, which was used to load the pre-trained **distilbert-base-uncased** model. This version of DistilBERT comprises 6 encoders, 12 bidirectional attention heads, 768 hidden size, and 66 million parameters and was used for all of the experiments with DistilBERT in this study.

The DistilBERT model training was conducted similarly to the BERT training, as detailed in section 3.6.1, using the same training arguments. The model evaluation results have been summarized below in Table 4.5.

| **Metric** | **Macro F1** | **Accuracy** | **Precision** | **Recall** |
|---|---|---|---|---|
| Mean | 73.17 | 78.27 | 74.10 | 72.75 |
| Standard Dev | 0.55 | 0.52 | 0.71 | 0.64 |
| Best Model | 74.15 | 79.18 | 75.37 | 73.68 |
| Worst Model | 72.05 | 77.37 | 73.00 | 71.67 |

**Table 4.5: DistilBERT Classification Average Performance**

The DistilBERT model underperformed compared to the BERT baseline on average, achieving a mean macro f1-score 0.88% lower than the BERT mean score.
As the research aims to optimize the DistilBERT fine-tuning process, a hyperparameter search was conducted to identify the optimal training parameters for future optimization experiments. The Optuna hyperparameter optimization framework was used to optimize the training process through efficient search strategies and early elimination of ineffective trials, enabling a larger number of trials within the resource constraints of the study. Forty trials were conducted across the parameter ranges defined in Table 4.6, using the HateTwitGab_1 dataset.

The parameters used in the best performing trial, as listed in Table 4.6, were used to train a newly created DistilBERT model to serve as the baseline. These training parameters differ in value from those used by the BERT model in the first experiment. The learning rate and the batch_size were reduced, while the number of warm-up steps and weight decay were added. The results of the DistilBERT baseline model performance are listed in table 4.7.

| Parameter | Range | Best Trial | BERT Training |
|---|---|---|---|
| **Epoch** | [2, 5] | 3 | 3 |
| **Batch Size** | [8,64] | 16 | 32 |
| **Learning Rate** | [1e-6, 1e-4] | 6.58e-5 | 1e-3 |
| **Seed** | [1,40] | 22 | 2 |
| **Weight Decay** | [0,0.3] | 0.289 | 0 |
| **Warmup Steps** | [0,500] | 464 | 0 |

**Table 4.6: DistilBERT Baseline Hyperparameter Search Results**

| Metric | Macro F1 | Accuracy | Precision | Recall |
|---|---|---|---|---|
| Mean | 75.87% | 80.30% | 76.44% | 75.79% |
| Standard Dev | 0.64% | 0.72% | 1.18% | 0.94% |
| Best Model | 76.99% | 81.51% | 78.39% | 76.08% |
| Worst Model | 74.92% | 79.84% | 76.10% | 74.92% |

**Table 4.7: DistilBERT Baseline Classification Results**

This baseline DistilBERT model using the configuration from the hyperparameter search outperforms the BERT baseline on average, achieving a mean macro f1-score 1.82% greater than the BERT mean score. The worst performing DistilBERT model managed to marginally outperform the best performing BERT model, achieving a macro f1-score of 74.92% compared to the BERT score of 74.89%.

| **Label** | **F1** | **Precision** | **Recall** |
|---|---|---|---|
| Hate Speech | 77.71% | 74.68% | 80.21% |
| Offensive | 87.12% | 86.53% | 87.84% |
| Normal | 63.32% | 68.12% | 59.31% |

**Table 4.8: DistilBERT Baseline Confusion Matrix**

The metrics for predicting the Hate Speech class label see the largest boost in performance over the BERT baseline. The macro F1 score increased by 3.3%, recall by 1.68%, and precision improved considerably by 3.96%. Performance in detecting offensive language increased marginally, with the F1 score, precision, and recall each achieving a gain of over 1 percent.

The validation loss for both the best and worst-performing models increases after the second epoch, diverging greatly from the training loss which indicates that overfitting is increasingly likely with subsequent epochs.

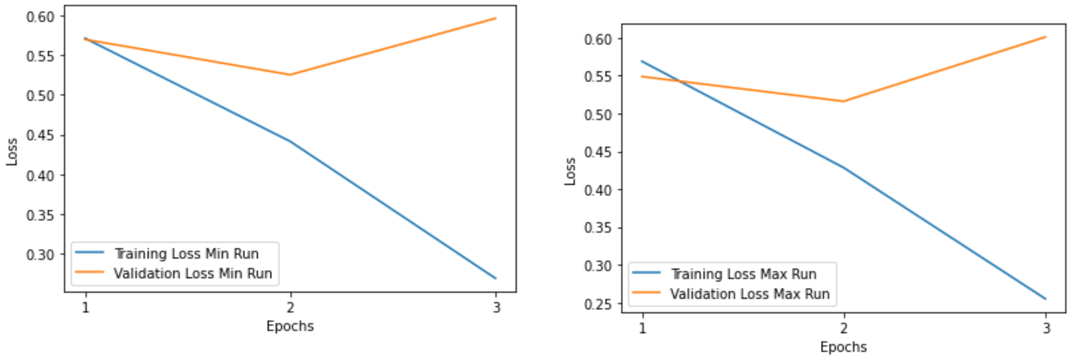


**Figure 4.3: DistilBERT Baseline Validation Loss**

## 4.4 Experiment 3: Weight Re-initialization

The purpose of this experiment was to evaluate the performance impact of the weight re-initializing of the DistilBERT model encoder layers prior to fine-tuning. Preliminary experiments were conducted using an instance of the *HateTwitGab* dataset to assess the optimal number of encoder layers to re-initialize. To re-initialize the weights across the layers, the *reinit_auto_encoder* method from the **stabilizer** python library was used. This algorithm required some minor modifications to handle the DistilBERT architecture.

This algorithm cycles through *n* layers and re-initializes the weights to a normal distribution with a mean of 0 and a standard deviation of 0.2, as described in the below pseudo code.

```
def reinit_autoencoder_model(model, n):
    """reinitialize autoencoder model layers"""
        for layer in model.transformer.layer[-n:]:
            for module in layer.modules():
                if (module.is_linear):
                  module.weight.data.normal_(mean=0.0, std=0.02)
    return model
```

A single training and evaluation run was conducted using an instance of the *HateTwitGab_1* dataset to set a local baseline prior to evaluating the impact of the weight re-initialization. Six subsequent trials were conducted which evaluated performance on the same dataset when re-initializing the weights for 1 to 6 layers. A summary of the results is listed below in Table 4.9.

| **Re-initialized Layers** | **Macro F1 %** | **Accuracy%** | **Precision%** | **Recall%** |
|---|---|---|---|---|
| ***0 (local baseline)*** | *76.00* | *80.66* | *77.43* | *74.94* |
| 1 | 74.92 | 79.30 | 75.00 | 75.19 |
| **2** | **75.30** | **79.81** | **76.11** | **74.85** |
| 3 | 75.25 | 79.86 | 74.90 | 75.05 |
| 4 | 75.00 | 79.58 | 75.52 | 75.00 |
| 5 | 74.12 | 78.72 | 74.95 | 74.18 |
| 6 | 72.13 | 77.60 | 74.14 | 71.36 |

**Table 4.9: Weight Re-initialization Layer Results**

The results illustrate that performance degrades through the re-initialization of layers. The f1-score recovers slightly after the top 2 layers have been initialized, however, performance continues to degrade if subsequent layers are re-initialized.

For the trials where weights were re-initialized, performance peaked when the two topmost layers were re-initialized. A hyperparameter search consisting of 40 trials was conducted to determine if the parameters identified during the hyperparameter search for the baseline needed to be adjusted for the modified model. The parameters for the best trial are shown in Table 4.10.

| **Parameter** | **Range** | **DistilBert Baseline** | **Weight Re-initialization Best Trial** |
|---|---|---|---|
| **Epoch** | [2, 5] | 3 | 3 |
| **Batch Size** | [8,64] | 16 | 16 |
| **Learning Rate** | [1e-6, 1e-4] | 6.58e-5 | 6.78e-5 |
| **Seed** | [1,40] | 22 | 31 |
| **Weight Decay** | [0,0.3] | 0.289 | 0.0595 |
| **Warmup Steps** | [0,500] | 464 | 491 |

**Table 4.10: Weight Re-initialization Hyperparameter Search Results**

To assess the performance impact more comprehensively, the average performance was evaluated against the 10 variations of the *HateTwitGab* dataset with two layers re-initialized. The results of the training and evaluations runs have been summarized below in Table 4.11.

| **Metric** | **Macro F1 %** | **Accuracy %** | **Precision %** | **Recall %** |
|---|---|---|---|---|
| Mean | 75.40 | 80.01 | 76.14 | 75.25 |
| Standard Dev | 0.99 | 0.66 | 0.90 | 1.38 |
| Best Model | 76.80 | 80.68 | 76.08 | 77.79 |
| Worst Model | 73.87 | 78.77 | 74.56 | 74.07 |

**Table 4.11: DistilBERT + Weight Re-initialization Classification Results**

The average performance of the DistilBERT model with the re-initialized weights outperforms the BERT baseline but underperforms against the DistilBERT baseline model.  The average macro F1 score, accuracy, precision, and recall underperform by 0.47, 0.29, 0.30, and 0.54 percent respectively when compared to the DistilBERT baseline.  The weight re-initialization introduces greater variance in the model results, with a standard deviation of 0.99 across the macro F1 score.

The metrics for predicting Hate Speech, Offensive and Normal see similar marginal declines in scores, with the exception of the precision of hate speech and offensive labels increasing by 0.06% and 0.46% respectively.

| **Label** | **F1** | **Precision** | **Recall** |
|---|---|---|---|
| Hate Speech | 77.03% | 74.74% | 79.71% |
| Offensive | 87.09% | 85.86% | 88.40% |
| Normal | 62.46% | 68.68% | 57.62% |

**Table 4.12: DistilBERT + Weight Re-initialization Confusion Matrix**

The validation loss follows a similar pattern to the DistilBERT baseline, with the validation loss increasing after the second epoch but with a greater divergence from the training loss.

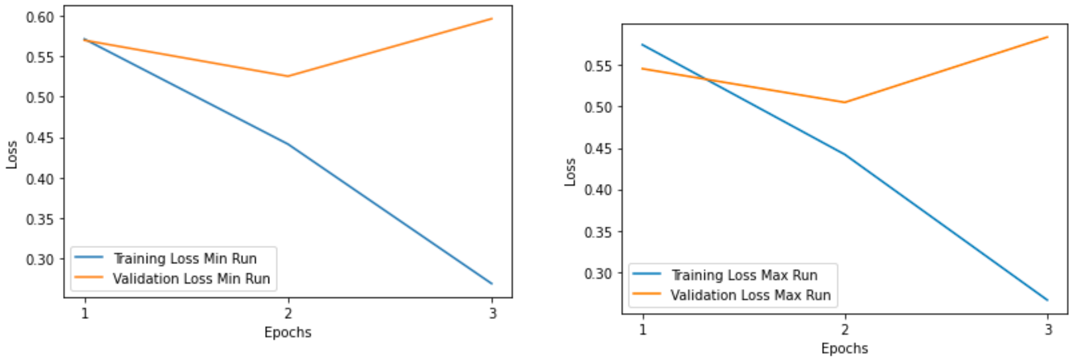


**Figure 4.4: DistilBERT + Weight Re-initialization Validation Loss**

## 4.5 Experiment 4: Intermediate Task Transfer

Performance improvements have been observed in pretrained models when training a model on a data-rich intermediate task. This experiment aims to evaluate if similar performance gains can be achieved with a DistilBERT model when fine-tuned on the hate speech classification task, using the SQuAD dataset as the intermediate task.

The SQuAD dataset was loaded via the Hugging Face datasets library. This dataset is pre-split into 90:10 ratio for training and validation but requires pre-processing before fine-tuning with the DistilBERT model for a question and answering task. Each row contains the text context and multiple related nested question and answer pairs. These required flattening which resulted in 87,599 question and answer pairs for training. Each question and its related context are tokenized as a combined pair for training inputs. As the combined length can exceed the default maximum sequence length of 512 for DistilBERT, the maximum length of the tokenized output was arbitrarily set to 384 to provide a compromise between truncation minimization and speed. The model requires the start and end positions of the answers as inputs into the training; therefore, an algorithm was developed to tokenize the answers and search for these within the context tokens.

A pretrained DistilBERT base model was used to fine-tune the processed SQuAD dataset using the **DistilbertForQuestionAnswering** model. As this represented a new task with a different dataset, the training parameters were reverted to those used in the BERT baseline model, as detailed below.

- Training Batch Size = 32
- Evaluation Batch Size = 32
- Epochs = 3
- Learning Rate = 1e-3
- Random Seed = 2

The fine-tuned model was saved, and the resulting model weights were loaded into the **DistilbertForSequenceClassification** model architecture. A hyperparameter search was conducted over 40 trials on a version of the *HateTwitGab* dataset, using the parameter ranges defined in Table 4.13. The resulting best-performing parameters are detailed in Table 4.13.

| Parameter | Range | DistilBert Baseline | Intermediate Task Best Trial |
|---|---|---|---|
| **Epoch** | [2, 3] | 3 | 3 |
| **Batch Size** | [16,32,64] | 16 | 64 |
| **Learning Rate** | [1e-6, 1e-4] | 6.58e-5 | 8.58e-5 |
| **Seed** | [1,40] | 22 | 32 |
| **Weight Decay** | [0,0.3] | 0.289 | 0.109 |
| **Warmup Steps** | [0,500] | 464 | 50 |

**Table 4.13: Intermediate Task Transfer Hyperparameter Search Results**

The best performing parameters were used to further fine-tune the SQuAD trained DistilBERT model against the 10 *HateTwitGab* datasets to evaluate the average performance, with the results recorded in Table 4.14.

| Metric | Macro F1 | Accuracy | Precision | Recall |
|---|---|---|---|---|
| Mean | 74.87 | 79.68 | 75.93 | 74.94 |
| Standard Dev | 0.74 | 0.68 | 1.35 | 1.12 |
| Best Model | 76.48 | 80.54 | 76.46 | 77.30 |
| Worst Model | 74.26 | 78.53 | 73.73 | 74.92 |

**Table 4.14: DistilBERT+ Intermediate Task Transfer Average Performance**

The average performance of the intermediate task experiments underperformed compared to the DistilBERT baseline model, with a macro F1 score 1% lower than the baseline.

| Label | F1 -Score | Precision | Recall |
|---|---|---|---|
| Hate Speech | 76.15 | 71.79 | 81.38 |
| Offensive | 86.79 | 86.11 | 87.56 |
| Normal | 61.68 | 69.89 | 55.88 |

**Table 4.15: DistilBERT + Intermediate Task Transfer Confusion Matrix**

The accuracy, precision, and recall are also lower than the baseline. The average class level F1 scores were outperformed by the baseline model, with only the recall for the hate speech and the precision of normal classes showing a marginal improvement over the baseline.

Analysis of the training and validation loss continues to demonstrate that fine-tuning for greater than two epochs results in an increase in validation loss and potentially overfitting.

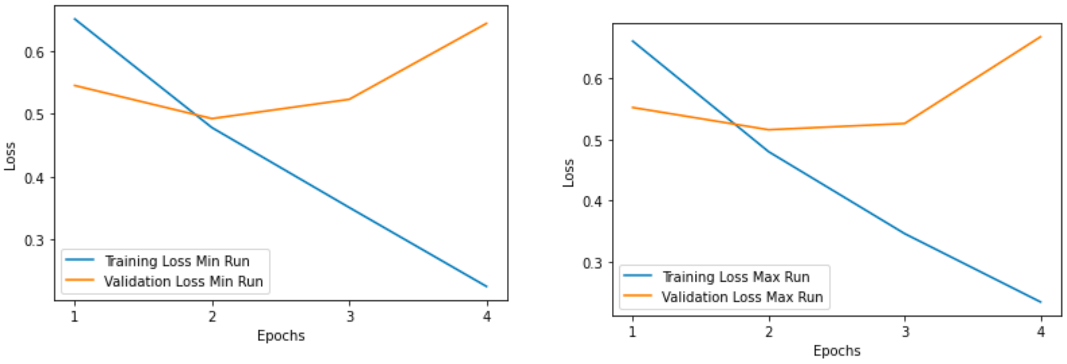


**Figure 4.5: DistilBERT + Intermediate Task Transfer Validation Loss**

## 4.6 Experiment 5: Layer-Wise Learning Rate Decay

The aim of this experiment was to determine if adjusting the learning rate per layer can improve the performance of the DistilBERT baseline model. Updated parameters need to be provided to the AdamW optimizer which includes the learning rate for each layer, with the learning decreasing, or decaying, with each descending layer from the topmost layer. To achieve this, an algorithm was developed based on the *get_optimizer_parameters_with_llrd* method from the **stabilizer** python library. The algorithm applies the peak learning rate to the topmost classification layer and applies a multiplier to the peak learning rate for each subsequent layer as described by the below formula.

$$Learning\ Rate\ =\ Peak\ Learning\ Rate\ *\ Multiplicative\ Factor^{layer\ number}$$

For example, the learning rate for the second transformer layer with a multiplicate factor of 0.95 and a peak learning rate of 1e-3 would be calculated as below.

$$Learning\ Rate\ =\ 1e-3\ *\ 0.95^2 =\ 0.0009025$$

To assess a suitable multiplicative factor to apply to the learning rate, preliminary experiments were conducted using an instance of the *HateTwitGab* dataset to assess the optimal factor from the range of [0.7, 0.75, 0.8,0.85,0.9, 0.95], using the learning rate of 6.58-e5, which is used by the DistilBERT baseline model. The results of these experiments are detailed in Table 4.16 below.

| Multiplier | Macro F1 | Accuracy | Precision | Recall | Hate F1 |
|---|---|---|---|---|---|
| 0.95 | **75.32** | **79.91** | **76.29** | **74.98** | 76.77 |
| 0.90 | 74.43 | 79.45 | 75.58 | 74.09 | **76.99** |
| 0.85 | 74.40 | 79.18 | 75.20 | 74.43 | 76.36 |
| 0.80 | 74.67 | 79.34 | 75.49 | 74.48 | 76.22 |
| 0.75 | 74.50 | 79.20 | 75.27 | 74.26 | 75.42 |
| 0.70 | 73.95 | 78.88 | 75.03 | 73.49 | 74.06 |

**Table 4.16: LLRD Multiplicative Factor Search Results**

The results indicate that the higher multiplier of 0.95, resulted in better F1 score, accuracy, precision, and performance in hate speech detection than in the other experiments. Decreasing the multiplier further and therefore increasing the learning rate decay results in a degradation in performance across all metrics. As models are evaluated using primarily the macro F1 score, the multiplicative factor of 0.95 was chosen as the optimal value.

A hyperparameter search was conducted using the selected multiplier value and parameter ranges detailed in Table 4.17 across 40 trials. The results of the hyperparameter search returned a learning rate of 2.12-e5 and a random seed of 7. Initial experiments resulted in a performance loss when compared to the trials conducted to determine the optimal learning rate decay. As result, the learning rate and random seed were adjusted to values used by the DistilBERT baseline.

| Parameter | Range | DistilBert Baseline | LLRD Parameters |
|---|---|---|---|
| **Epoch** | [2, 3] | 3 | 3 |
| **Batch Size** | [16,32,64] | 16 | 16 |
| **Learning Rate** | [1e-6,1e-4] | 6.58e-5 | 6.58e-5 |
| **Seed** | [22] | 22 | 22 |
| **Weight Decay** | [0,0.3] | 0.289 | 0.129 |
| **Warmup Steps** | [0,500] | 464 | 164 |

**Table 4.17: LLRD Hyperparameter Search Results**

The values defined in Table 4.17 and a learning rate decay multiplier of 0.95 were used to evaluate the average performance across the 10 variations of the *HateTwitGab* datasets, with the results summarized in Table 4.18 below.

| Metric | Macro F1 | Accuracy | Precision | Recall |
|---|---|---|---|---|
| Mean | 75.95 | 80.33 | 76.34 | 76.00 |
| Standard Dev | 0.42 | 0.39 | 0.69 | 1.06 |
| Max | 76.13 | 80.51 | 76.65 | 77.86 |
| Min | 75.38 | 80.07 | 75.85 | 75.13 |

**Table 4.18: DistilBERT + LLRD Average Classification Results**

The average performance of the DistilBERT model with the LLRD of 0.95 marginally outperforms the DistilBERT baseline model. The average macro f1 score, accuracy, and recall outperforms by 0.08, 0.03, and 0.21 percent respectively when compared to the DistilBERT baseline, while precision underperforms by 0.1%.

| Label | F1 | Precision | Recall |
|---|---|---|---|
| Hate Speech | 77.44% | 74.44% | 80.97% |
| Offensive | 87.10% | 86.72% | 87.56% |
| Normal | 63.31% | 67.89% | 59.44% |

**Table 4.19: DistilBERT + LLRD Confusion Matrix**

The validation loss follows a similar pattern to the DistilBERT baseline, with the validation loss increasing after the second epoch but with a greater divergence from the training loss.

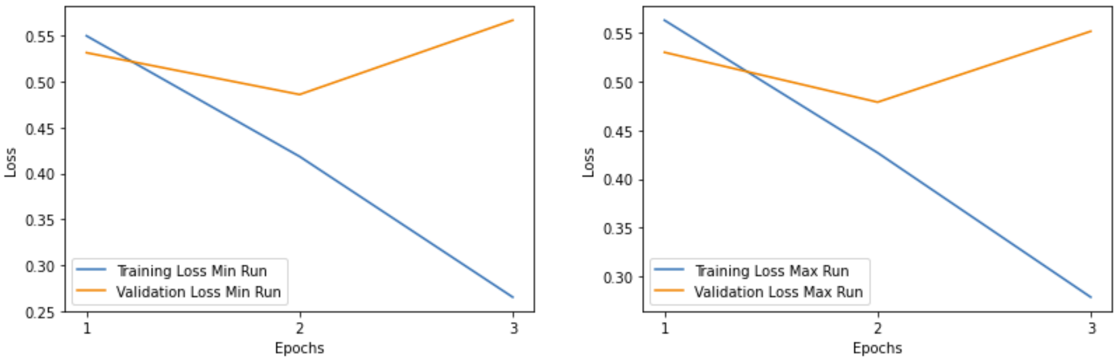


**Figure 4.6: DistilBERT + LLRD Validation Loss**

## 4.7 Experiment 6: Optimal Model

In addition to evaluating the optimization strategies independently, further experiments were conducted to evaluate the performance impact of combining the weight re-initialization (WR), layer-wise learning rate decay (LLRD) and the intermediate task transfer (ITT) strategies. An initial assessment was performed on each combination, fine-tuning the DistilBERT models on the HateTwitGab_1 dataset using the same training arguments as the baseline DistilBERT model. Each combination was trained for three epochs. The configuration values used in the best performing models in the weight re-initialization and LLRD experiments were applied for this experiment. For weight re-initialization, the two topmost encoder layers were re-initialized. For LLRD, the multiplicative factor was set to 0.95. The results of these experiments are listed in Table 4.20.

| **Model** | **Macro F1** | **Accuracy** | **Precision** | **Recall** |
|---|---|---|---|---|
| ITT + WR + LLRD | 74.44 | 79.30 | 75.28 | 74.30 |
| ITT + LLRD | 75.23 | 79.78 | 75.23 | 75.09 |
| ITT + WR | 75.83 | 80.36 | 75.84 | **75.90** |
| WR +LLRD | **75.86** | **80.38** | **75.86** | 75.14 |

**Table 4.20: Fine-Tuning Strategy Combination Performance**

The combination of all three optimization strategies achieved the lowest macro F1 score of 74.44. The weight re-initialization in combination with either ITT (75.83) or LLRD (75.86) both outperformed the average performance of the DistilBERT baseline model. The combination of the weight re-initialization and LLRD edged performance based on the macro F1 score and was thus selected as the optimal model configuration to be advanced for further experimentation.

A hyperparameter search consisting of 40 trials was conducted using the *HateTwitGab*_1 dataset to evaluate if further performance gains can be achieved through optimizing the training parameters. The search parameters were reduced in range based on the findings in earlier experiments, except for the batch size range, which was extended to include a size of 128.

| **Parameter** | **Range** | **DistilBert Baseline** | **WR +LLRD Parameters** |
|---|---|---|---|
| **Epoch** | [3] | 3 | 3 |
| **Batch Size** | [16,32,64,128] | 16 | 128 |
| **Learning Rate** | [1e-6,1e-4] | 6.58e-5 | 10.58e-5 |
| **Seed** | [22] | 22 | 22 |
| **Weight Decay** | [0,0.3] | 0.289 | 0.289 |
| **Warmup Steps** | [0,500] | 464 | 464 |

**Table 4.21: WR+ LLRD Model Hyperparameter Search Results**

Previous experiments had observed a loss convergence after one or two epochs, with validation loss increasing after the 2$^{nd}$ epoch. A larger batch size can reduce the effects of overfitting; therefore, the higher batch size was added to evaluate its effects. Details of the modified parameter ranges and best performing trial configuration can be found in Table 4.21. Consistent with previous research, this configuration was used to evaluate the average performance across the 10 variations of the *HateTwitGab* datasets, with the results summarized in Table 4.22 below.

| **Metric** | **Macro F1** | **Accuracy** | **Precision** | **Recall** |
|---|---|---|---|---|
| Mean | 76.13 | 80.51 | 76.51 | 76.15 |
| Standard Dev | 1.14 | 0.83 | 0.76 | 1.31 |
| Best Model | 77.47 | 81.60 | 77.65 | 77.66 |
| Worst Model | 73.48 | 78.83 | 75.81 | 72.94 |

**Table 4.22: DistilBERT + WR + LLRD Average Classification Results**

The average performance of the DistilBERT model, using LLRD and weight re-initialization, outperforms the DistilBERT baseline model's macro average F1 score by 0.26%. Performance gains were achieved across accuracy (+0.21%), precision (+0.06%), and recall (+0.36). Compared to the BERT baseline, the optimized DistilBERT model registered improvements across the macro average F1 (+ 2.12%), accuracy (+1.74 %), precision (+2.05%) and recall (+2.13%).

The optimized model has a higher variance than the other models generated, with a standard deviation of 1.14 across the macro F1-score. This model managed to achieve both the highest (77.47%) and lowest (73.48%) macro F1 score during the average performance testing across all DistilBERT experiments, with only the worst-performing BERT model registering a lower macro F1 score (73.28%). The high standard deviation matches similar observations when testing the weight re-initialization independent of the LLRD, suggesting this may cause instability.

| **Label** | **F1** | **Precision** | **Recall** |
|---|---|---|---|
| Hate Speech | 78.60 | 75.36 | 82.12 |
| Offensive | 87.17 | 86.60 | 87.76 |
| Normal | 62.62 | 67.56 | 58.55 |

**Table 4.23: DistilBERT + WR + LLRD Confusion Matrix**

The optimized model achieved performance gains in the macro F1 score for hate speech (+0.89%) and offensive language detection (0.05%) but registered a decrease in performance when it comes to the classification of the normal label (-0.60%).

## 4.8 Evaluation

A summary of the experiments conducted in this research can be found in Table 4.24, which provides evidence that optimizing the fine-tuning process on pretrained DistilBERT models can result in performance gains over a vanilla BERT model. The DistilBERT baseline, DistilBERT LLRD and DistilBERT WR+LLRD offered the best performance in comparison to the BERT baseline, achieving gains in the average average macro F1 score ranging from 1.82 to 2.07%.

| Model | Macro F1 | Accuracy | Precision | Recall | Hate-F1 % |
|---|---|---|---|---|---|
| BERT Baseline | 74.05 | 78.77 | 74.46 | 74.09 | 74.41 |
| DistilBERT Baseline | 75.87 | 80.30 | 76.44 | 75.79 | 77.18 |
| DistilBERT WR | 75.53 | 80.01 | 76.42 | 75.25 | 77.03 |
| DistilBERT LLRD | 75.95 | 80.33 | 76.34 | 76.00 | 77.44 |
| DistilBERT ITT | 74.87 | 79.68 | 75.93 | 74.94 | 76.15 |
| DistilBERT + WR +LLRD | **76.13** | **80.51** | **76.51** | **76.15** | 78.60 |

**Table 4.24: Summary of models performance**

To determine the statistical test required to validate the significance of the performance gains, the results of the models performance over the 10 variations of the *HateTwitGab* dataset were tested for normality.

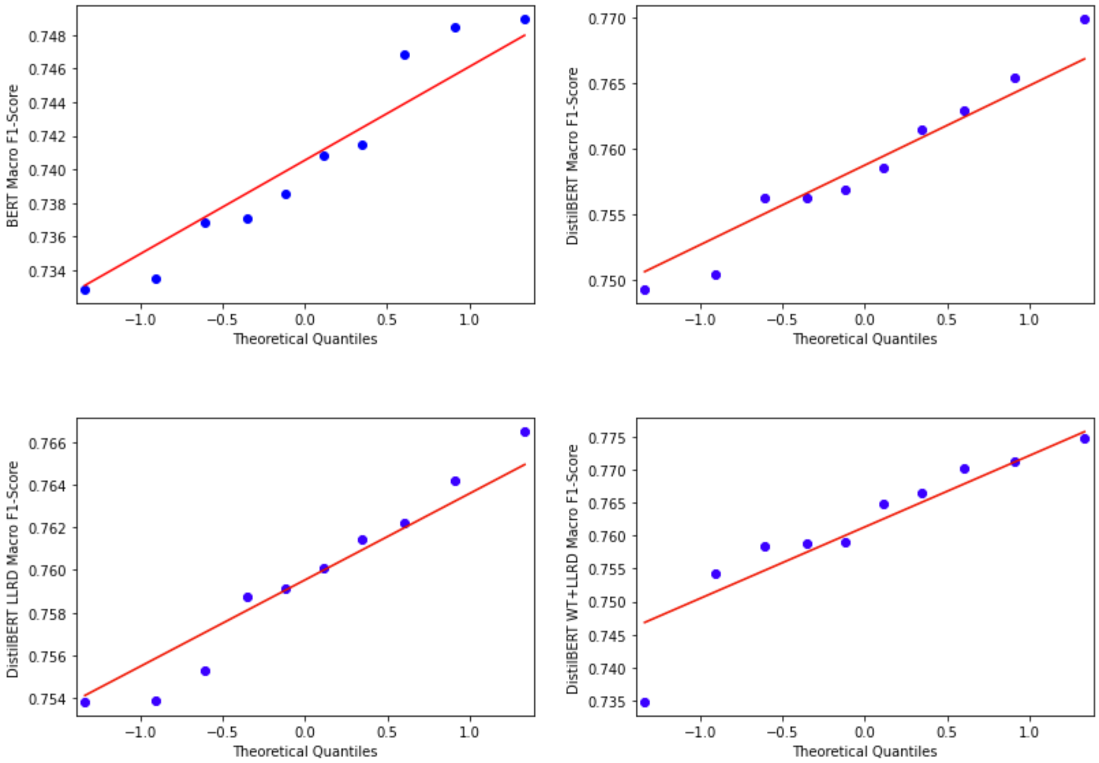


**Figure 4.7: Model Q-Q Scatterplots**

Q-Q plots were generated to visually test the macro F1 score performance distribution for normality. The scatterplot charts illustrated in Figure 4.7 highlight that each model's performance resembles a normal distribution. Given the small sample sizes, Shapiro-Wilk tests were conducted to test the null hypothesis that the macro F1 scores for each model were drawn from a normal distribution. The results documented in Table 4.25 did not show evidence of non-normality, resulting in the acceptance of the null hypothesis of normal distribution.

| Model | W (Test Statistic) | P Value |
|---|---|---|
| BERT Baseline | 0.92 | 0.24 |
| DistilBERT Baseline | 0.97 | 0.88 |
| DistilBERT + LLRD | 0.95 | 0.68 |
| DistilBERT +WR +LLRD | 0.89 | 0.16 |

**Table 4.25: Tests for normality**

A one-tailed paired t-test was selected to accept the below hypothesis. This statistical test was chosen due to the normality of the data, the same training data being used across models, and the need to test for performance improvement at a 0.05 alpha level instead of just confirming a significant difference in results.

***Null Hypothesis:*** *The pre-trained BERT base model statistically outperforms a DistilBERT model, which employs fine-tuning optimization strategies, on the macro average F1 score when fine-tuned on the social media dataset for the target task of hate speech classification.*

There was significant increase in the macro average F1 score with the DistilBERT + WR + LLRD model (M=76.13, SD=1.14) compared to the BERT baseline mode (M=74.05, SD=0.59), $t(9)=-6.25, p < 0.001$. The DistilBERT + LLRD (M= 75.95, SD=0.42) also reported a significant performance increase in the average macro F1 score compared to the BERT baseline (M=74.05, SD=0.59), $t(9)=-8.54, p < 0.001$. As the best-performing optimized models significantly outperformed the BERT baseline, there is sufficient evidence to reject the null hypothesis.

The baseline DistilBERT model (M=75.87, SD=0.64) also significantly outperformed the average macro F1 score when compared to the BERT baseline (M=74.05, SD=0.59), *t*(9)=-7.31, $p < 0.001$. This suggests that there are opportunities for significant performance gains over the BERT performance through hyperparameter tuning alone.

The experiments corroborated the authors of the DistilBERT research papers' findings that the model performed faster than BERT (Sanh et al.,2020). The average evaluation time for the DistilBERT baseline model (3.92 secs) was 35% faster than the BERT baseline (6.09 secs), with the DistilBERT+WR+LLRT evaluating 38% faster (3.78 secs). BERT models were also significantly larger than the DistilBERT models, with the best performing fine-tuned model consuming 417.7 MB storage compared with 255.5MB for the best performing DistilBERT+WR+LLRD model.

An evaluation of the BERT baseline model (BERT BL) , DistilBERT model (DistilBERT BL), and the optimal DistilBERT WR +LLRD (DistilBERT OP) performance against the class labels can be found in below Table 4.26.

| **Class Label** | **Model** | **F1 %** | **Precision %** | **Recall %** |
|---|---|---|---|---|
| **Hate Speech** | *BERT BL* | 74.40 | 70.50 | 78.61 |
| | *DistilBERT BL* | 77.71 | 74.68 | 80.21 |
| | *DistilBERT OP* | 78.60 | 75.36 | 82.12 |
| **Offensive** | *BERT BL* | 86.08 | 85.58 | 86.59 |
| | *DistilBERT BL* | 87.12 | 86.53 | 87.84 |
| | *DistilBERT OP* | 87.17 | 86.60 | 87.76 |
| **Normal** | *BERT BL* | 61.68 | 67.17 | 57.07 |
| | *DistilBERT BL* | 63.32 | 68.12 | 59.31 |
| | *DistilBERT OP* | 62.62 | 67.56 | 58.55 |

**Table 4.26: Summary of Class Label Results**

The DistilBERT baseline and DistilBERT WR+LLRD offer the most significant F1 performance gains in hate speech detection than other class labels compared to the BERT baseline, achieving gains of 3.31% and 4.20%, respectively. The DistilBERT models registered more modest gains in the F1 scores for the offensive and normal class labels, achieving gains of roughly 1%. The DistilBERT WR +LLRD model handled the offensive language (87.17%) best, unsurprising given the dominance of the class label (53%) in the training sets. This model also achieved a respectable F1 score of 78.60% in hate speech detection.

Unlike previous research observations, which identified the conflation of hate speech and offensive language as significant challenges, the DistilBERT WR + LLRD model's overall performance was impacted by the poor performance in the classification of normal data. Similar poor performance in the classification of normal data was observed across all experiments, suggesting that there may be some issues with the underlying data set warranting further analysis.

## 4.9 Discussion

The primary objective of this study was to determine if the more compact, faster DistilBERT model could outperform a vanilla BERT base model by employing strategies to optimize the model fine-tuning. In section 1. 2, the null hypothesis was developed to assume that a BERT model would outperform the DistilBERT models generated using fine-tuning optimization strategies. This research tested this hypothesis by evaluating a series of sub-questions. This section will evaluate these questions in relation to the results of the experiments conducted and conclude with a review of the evidence to reject the null hypothesis,

For the research sub-question A, the question was posed to determine, ***"Does a standard BERT model outperform a DistilBERT model in the task of hate speech classification ?"*** The DistilBERT author's observations that the model can achieve comparable results yet still underperform compared to BERT held true when testing on the hate speech classification task. The initial DistilBERT model tested achieved 98.8% of the performance of the BERT base, with an average macro f1 score 0.88% lower than the 74.05% achieved by the BERT model. After a hyperparameter search which included adjusting the learning rate, batch size, random seed, and weight decay, the DistilBERT managed to significantly outperform the BERT model by 1.86%.

To evaluate the impact of the fine-tuning strategies, the research sub-question B asked ***"Does the re-initializing of the weights of layers in a DistilBERT model before fine-***

***tuning improve its performance?”.*** The experiments showed that the weight re-initialization had an adverse effect on performance when applied to a DistilBERT model. Initial tests suggested that re-initializing the two topmost layers gave the best performance from the range of one to six layers but ultimately degraded performance. This was corroborated by the average macro f1score underperforming the DistilBERT baseline by 0.47%. The experiments also highlighted additional variance in results when weights are re-initialized, increasing the standard deviation of the macro f1 score from 0.66 to 0.99 compared to the DistilBERT baseline. The weight re-initialization may prove more impactful on large models with more layers, such as BERT, which has twelve encoder layers versus DistilBERT’s six. The topmost layers in BERT may be more overly-specialized for the pre-training objective than DistilBERT. The smaller number of layers in the DistilBERT model may lead to catastrophic forgetting when the weights are re-initialized.

To determine the performance impact of applying LLRD to hate speech classification, research sub-question C was formulated as ***“Does the application of LLRD in a DistilBERT model improve its performance?”.*** The experiments showed that applying an LLRD rate of 0.95 proved marginally beneficial to performance, increasing performance compared to the DistilBERT baseline by 0.08%. The model performance was more stable across the trials, recording a standard deviation of 0.42 compared to 0.59 and 0.64, respectively for the BERT and DistilBERT baseline models.

The experiments conducted showed a performance degradation when evaluating the research sub-question ***“Does fine-tuning a DistilBERT model on an intermediate task before fine-tuning on the hate speech classification task improve its performance?”.*** Fine-tuning the model on the SQuAD dataset before fine-tuning on the *HateTwitGab* dataset yielded the worst performing DistilBERT model from the optimization strategies. While the model still outperformed the BERT model, it underperformed the DistilBERT baseline average macro f1-score by 1%. There may be other intermediate tasks such as the Cosmos QA and HellaSwag that may yield better results. However, this research found the intermediate task transfers to be counterproductive to performance when applied independently and in combination with either LLRD or weight re-initialization.

The evaluation of research sub-question E, ***"Which combination of weight re-initialization, LLRD, and intermediate task transfers with a DistilBERT model results in the best performance?"***, elected weight re-initialization and LLRD as the optimal combination during initial trial comparisons. The combination of these strategies achieved the highest average macro F1 score (76.13%) throughout the experiments conducted in this study, outperforming both the DistilBERT and BERT baselines.

The other permutations of strategy combinations yielded surprising results. The combination of LLRD and ITT proved counterproductive as trials with this pair resulted in the lowest results, with the combination of ITT, LLRD, and WR achieving the weakest results. The weight re-initialization and intermediate task transfer combined achieved the second-highest macro F1 score at 75.83% during the trial comparisons. The combination of both outperformed the average macro F1 scores achieved in their independent tests. This would suggest that the intermediate task transfer may provide greater benefit when adjusting the weights in the lower layers for more general feature learning and may adversely skew weights in the higher layers for more task-specific learning.

The previous research questions provided the answers to assess the final research sub-question F, "***Do any of the fine-tuning optimisation strategies result in a DistilBERT model that significantly outperforms a standard BERT model in the task of hate speech classification of social media data?".*** A hyperparameter search was sufficient to identify an optimal DistilBERT model configuration which outperformed the average macro F1 performance of the BERT base model in hate speech classification when fine-tuned on the *HateTwitGab* dataset. The research conducted identified that the employment of LLRD is further advantageous to the macro F1 performance of the DistilBERT model in the classification of hate speech data, with further performance gains achieved when combined with re-initializing the weights of the two topmost encoder layers in the model.

### 4.9.1 Strengths of Results

- The evidence provided through the experiments conducted during this research confirms the hypothesis that a DistilBERT model can outperform the large BERT base model in hate speech classification through the application of fine-tuning optimization techniques such as LLRD and weight re-initialization.
- The DistilBERT model utilizing weight re-initialization and LLRD during fine-tuning improves the average F1 score in hate speech classification by over 4% to 78.6% compared to the BERT model. The model offers strong and comparable performance to BERT in identifying offensive content, achieving an F1 score of 87.17%.
- The research shows that a DistilBERT model can offer a viable alternative to larger BERT models in terms of performance, and is up to 38% faster and over 60% smaller in size. This can improve performance speed and scalability, while reduce both the financial costs and environmental impact.

### 4.9.2 Limitations of Results

- Significant performance gains in the macro F1 score were achieved through the DistilBERT model hyperparameters identified by a hyperparameter search. While hyperparameter searches are necessary to identify the complementary configuration to maximize the results of the optimization strategies employed, further work could be performed to individually assess the impact of batch size, weight decay, and warm up steps in each of the experiments.
- The performance gains achieved through the fine-tuning strategies and hyperparameter searches employed may provide similar performance gains when applied to the BERT model. The computational costs and training times were too prohibitive to assess within this research.
- The initial tests used to identify configuration settings for each optimization strategy were only evaluated against a single iteration of the *HateTwitGab* dataset due to resource and time constraints. Evaluating each configuration permutation against the ten variations of the *HateTwitGab* dataset may result in other configurations achieving a higher average macro F1 score than the values selected for further analysis.

- The experiments conducted to assess the performance impact of intermediate task transfers were limited by selecting a single intermediate task for evaluation. A broader assessment of intermediate task types would be necessary to develop a conclusive understanding of the performance benefits.

# 5 CONCLUSION

This chapter aims to summarize the thesis, providing a concise overview of the research, problem definition, the experiments conducted, and the findings. The chapter will expand upon the lessons learned from the research into optimizing DistilBERT models to compete with larger transformer architectures and the contributions to the sub-domain of hate speech classification in social media. This section will conclude with a discussion on the opportunities for further work based on this research.

## 5.1 Research Overview

This research aimed to evaluate the hypothesis that a DistilBERT model could outperform the larger BERT base model in the classification of hate speech in social media, through the employment of the following fine-tuning strategies: weight re-initialization, layer-wise learning rate decay, and intermediate task transfers.

## 5.2 Problem Definition

The proliferation of hate speech in online social media has increased media attention due to its profound impact on its victims' psychological and physical safety. Social media platforms and governments have attempted to deter this through usage policies, moderation, and legislation that prosecutes the perpetrators, albeit with limited success.

The research community has sought to help tackle this issue by developing benchmark datasets and models to automate hate speech detection, with recent research gravitating towards large pre-trained transformer-based architectures such as BERT since their emergence as state of the art across a variety of NLP tasks. These ever-growing transformer models have raised concerns around their size, speed, and environmental costs, leading to the research detailed in chapter 2 that proposes smaller, lighter weight models such as DistilBERT, Tiny BERT, and MobileBERT. DistilBERT offers a lighter, faster alternative but comes with a performance penalty that could be prohibitive to its adoption in social media.

This research seeks to evaluate if fine-tuning strategies successfully employed on larger models can also be applied to DistilBERT to offer more competitive performance in the domain of hate speech detection in comparison to BERT, leading to the research question: *"Can a smaller, faster transformer model such as DistilBERT outperform a standard BERT model in the task of hate speech classification of social media data through the application of weight re-initialization, layer-wise linear rate decay and intermediate task transfers during fine-tuning?"*

The DistilBERT model achieved 97% of the performance of BERT in the original paper (Sahn et al., 2019). This lead to the formulation of the null hypothesis documented in section 1.3 that assumed a BERT base model will outperform a DistilBERT model that employs the fine-tuning strategies listed in the research question. The acceptance of the alternate hypothesis that DistilBERT can achieve greater performance would propose a cheaper, smaller, faster, and more environmentally friendly alternative to the ever-growing and slower large transformer models to help combat hate speech in social media.

## 5.3 Experimentation, Evaluation and Results

To test the hypothesis formulated, a dataset was generated by combining two datasets generated from previous research (Davidson et al., 2017; Mathew et al., 2021) comprising labelled data from the social media networks Twitter and Gab. This combined dataset is referred to as the *HateTwitGab* dataset throughout the research. Initial experiments were conducted to establish baselines in performance, fine-tuning the vanilla BERT base and DistilBERT models ten times on shuffled variations of the *HateTwitGab* dataset to calculate average performance metrics using stratified 8:1:1 splits for training, validation, and testing. The BERT base model achieved a higher average macro F1 score (74.05%) than the vanilla DistilBERT model (73.17%). A hyperparameter search on the vanilla DistilBERT model identified configuration that achieved a gain of 2.7% in the average macro F1 score (75.87%). The results were used as the DistilBERT baseline for comparison.

To assess the impact of the weight re-initialization, LLRD, and intermediate task transfer fine-tuning strategies, independent experiments were conducted that employed each strategy to the DistilBERT model. Experiments that applied the weight re-initialization across the DistilBERT model layers found that applying to the two topmost layers offered the best results but ultimately underperformed when compared to the DistilBERT baseline. This could be due to the strategy performing better with smaller datasets to enable significant adjustment of the weights to the downstream task. For the intermediate task transfers, experiments were performed that fine-tuned a DistilBERT model on the SQuAD questioning and answering task before fine-tuning on the hate speech classification with the *HateTwitGab* dataset. These experiments showed degradation in the macro F1 score (-0.34%) in comparison to the DistilBERT baseline, achieving the lowest result from the independent strategy tests. For LLRD, the experiments performed when fine-tuning the DistilBERT model showed a marginal performance gain over the baseline in the average macro F1 score (+0.08%) when applying an LLRD of 0.95 with a learning rate of 6.58e-5.

The compound effect of the three fine-tuning strategies was assessed by conducting experiments to compare the performance of their four combinations. The combination of all three strategies resulted in the lowest macro F1 score. The combination of the weight re-initialization and LLRD provided the best results from the combined strategies, achieving the best average macro F1-score (76.13%) from the experiments conducted, outperforming both the BERT (+2.08%) and DistilBERT baselines (+0.26%). This optimized model achieved a 4% F1 score improvement in the classification of the hate speech label compared to the BERT baseline while registering smaller improvements in the classification of the offensive (+1.09%) and normal labels (+0.94%).

One-tailed paired t-test demonstrated a significant increase in the average macro F1-score of the DistilBERT model with weight re-initialization and LLRD applied (M=76.13, SD=1.14) compared to the BERT baseline model (M=74.05, SD=0.59), $t(9)=-6.25$, $p < 0.001$, providing evidence to reject the null hypothesis and affirmatively answer the research question posed.

## 5.4 Contributions and Impact

Research in the NLP domains has seen performance gains by employing ever-increasing language models. Research into the automation of hate speech detection has gravitated towards these larger models to achieve state-of-the-art performance on common hate speech benchmark datasets,  despite concerns regarding the scalability, costs, and environmental impact of their adoption on large social media networks.

This research provides empirical evidence that a DistilBERT model can outperform a BERT base model in the classification of hate speech through the employment of fine-tuning strategies.  LLRD offers performance benefits when applied to the smaller DistilBERT model, performance gains are increased when combined with weight re-initialization.
The findings in this research provide alternative models and fine-tuning strategies to the academic research community over BERT.  The combination of competitive performance with lower costs and faster training times offers a more accessible option to the broader research community, which is conducive to extensive experimentation.

Furthermore, this research presents the pre-processed and labelled *HateTwitGab* dataset for future research.  This dataset provides a larger, more balanced distribution of class labels and targeted groups across the Twitter and Gab social media platforms than the original datasets which it is derived from.

## 5.5 Future Work and Recommendations

The application of lighter, cheaper, faster models to the problem of hate speech detection is an avenue that warrants further research due to the economic and environmental benefits.  This research focused on the DistilBERT, but other lightweight models generated through knowledge distillation, such as TinyBERT and MobileBERT, could provide superior performance.  The application of fine-tuning strategies such as weight re-initialization and LLRD could benefit MobileBERT due to the depth of its 24 layers.  Emerging research has produced models such as Linformer, which claims significant time and memory savings, and comparable performance with larger transformer models.   Limited peer-reviewed research and support for this model currently exist; however, its employment by Facebook and Instagram to tackle hate speech on their platforms would justify further investment by the research community to compare its performance to the state-of-the-art in hate speech detection.

With the DistilBERT models evaluated through this research,  there are opportunities to analyse to isolate and assess the impact of the different hyperparameter settings for weight decay, batch size and warm-up steps on classification performance.  This research would suggest that analysis of further hyperparameter testing, such as modifying dropout to increase regularization and reduce the overfitting identified during the experiments conducted, should be investigated to determine the impact on performance.

The models produced during this research performed poorly on the classification of the normal label in comparison to the hate speech and offensive labels in the *HateTwitGab* dataset.  A more comprehensive analysis of the underperformance may help discover additional preprocessing requirements or sampling strategies for the dataset that could improve the classification of normal data and overall model macro F1 score.  The models through this research could be evaluated against other benchmark hate speech datasets, such as those listed in the systematic review of benchmark corpora by Poletto et al. (2020), to determine the generalizability of models fine-tuned on the *HateTwitGab* dataset and the value of this dataset towards further research in this field.

# BIBLIOGRAPHY

Awan, I., & Zempi, I. (2015). *We fear for our lives: Offline and online experiences of anti-Muslim hostility*.

Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep Learning for Hate Speech Detection in Tweets. *Proceedings of the 26th International Conference on World Wide Web Companion*, 759–760. https://doi.org/10.1145/3041021.3054223

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T. J., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. *ArXiv*, *abs/2005.14165*.

Burnap, P., & Williams, M. (2015). Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making: Machine Classification of Cyber Hate Speech. *Policy & Internet*, *7*. https://doi.org/10.1002/poi3.85

Burnap, P., & Williams, M. L. (2016). Us and them: Identifying cyber hate on Twitter across multiple protected characteristics. *EPJ Data Science*, *5*(1), 11. https://doi.org/10.1140/epjds/s13688-016-0072-6

Cao, R., Lee, R. K.-W., & Hoang, T.-A. (2020). DeepHate: Hate Speech Detection via Multi-Faceted Text Representations. *12th ACM Conference on Web Science*, 11–20. https://doi.org/10.1145/3394231.3397890

Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., & Vakali, A. (2017). Mean Birds: Detecting Aggression and Bullying on Twitter. *Proceedings of* *the 2017 ACM on Web Science Conference*, 13–22. https://doi.org/10.1145/3091478.3091487

Chen, Y., Zhou, Y., Zhu, S., & Xu, H. (2012). Detecting Offensive Language in Social Media to Protect Adolescent Online Safety. *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, 71–80.

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *ArXiv*, *abs/2003.10555*.

Davidson, T., Bhattacharya, D., & Weber, I. (2019). Racial Bias in Hate Speech and Abusive Language Detection Datasets. *Proceedings of the Third Workshop on Abusive Language Online*, 25–35. https://doi.org/10.18653/v1/W19-3504

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). *Automated Hate Speech Detection and the Problem of Offensive Language*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *CVPR*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., & Bhamidipati, N. (2015). *Hate Speech Detection with Comment Embeddings* (p. 30). https://doi.org/10.1145/2740908.2742760

Dwivedi, Y. K., Ismagilova, E., Hughes, D. L., Carlson, J., Filieri, R., Jacobson, J., Jain, V., Karjaluoto, H., Kefi, H., Krishen, A. S., Kumar, V., Rahman, M. M., Raman, R., Rauschnabel, P. A., Rowley, J., Salo, J., Tran, G. A., & Wang, Y. (2021). Setting the future of digital and social media marketing research: Perspectives and research propositions. *International Journal of Information Management*, *59*, 102168. https://doi.org/10.1016/j.ijinfomgt.2020.102168

Fortuna, P., & Nunes, S. (2018). A Survey on Automatic Detection of Hate Speech in Text. *ACM Comput. Surv.*, *51*(4). https://doi.org/10.1145/3232676

Gambäck, B., & Sikdar, U. K. (2017). Using Convolutional Neural Networks to Classify Hate-Speech. *Proceedings of the First Workshop on Abusive Language Online*, 85–90. https://doi.org/10.18653/v1/W17-3013

Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., & Plagianakos, V. P. (2018). Convolutional Neural Networks for Toxic Comment Classification. *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*. https://doi.org/10.1145/3200947.3208069

Gewirtz, P. (1996). On 'I Know It When I See It'. *The Yale Law Journal*, *105*(4), 1023–1047.

Greevy, E., & Smeaton, A. F. (2004). Classifying Racist Texts Using a Support Vector Machine. *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 468–469. https://doi.org/10.1145/1008992.1009074

Gröndahl, T., Pajola, L., Juuti, M., Conti, M., & Asokan, N. (2018). All You Need is 'Love': Evading Hate Speech Detection. *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*.

Hawdon, J., Oksanen, A., & Räsänen, P. (2017). Exposure to Online Hate in Four Nations: A Cross-National Consideration. *Deviant Behavior*, *38*(3), 254–266. https://doi.org/10.1080/01639625.2016.1196985

He, P., Liu, X., Gao, J., & Chen, W. (2021). DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *ArXiv*, *abs/2006.03654*.

Howard, J., & Ruder, S. (2018). *Universal Language Model Fine-tuning for Text Classification*. 328–339. https://doi.org/10.18653/v1/P18-1031

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751. https://doi.org/10.3115/v1/D14-1181

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*, 84–90.

Kumar, R., Lahiri, B., Ojha, A. K., & Bansal, A. (2020). ComMA@FIRE 2020: Exploring Multilingual Joint Training across different Classification Tasks. *FIRE*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, *abs/1907.11692*.

Matamoros-Fernández, A., & Farkas, J. (2021). Racism, Hate Speech, and Social Media: A Systematic Review and Critique. *Television & New Media*, *22*, 205–224.

Mathew, B., Dutt, R., Goyal, P., & Mukherjee, A. (2019). Spread of Hate Speech in Online Social Media. *Proceedings of the 10th ACM Conference on Web Science*, 173–182. https://doi.org/10.1145/3292522.3326034

Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., & Mukherjee, A. (2021). HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(17), 14867–14875.

Mehdad, Y., & Tetreault, J. (2016). *Do Characters Abuse More Than Words?* (p. 303). https://doi.org/10.18653/v1/W16-3638

Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ICLR*.

Mollas, I., Chrysopoulou, Z., Karlos, S., & Tsoumakas, G. (2022). ETHOS: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*, 1–16.

Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media. In H. Cherifi, S. Gaito, J. F. Mendes, E. Moro, & L. M. Rocha (Eds.), *Complex Networks and Their Applications VIII* (pp. 928–940). Springer International Publishing.

Müller, K., & Schwarz, C. (2020). Fanning the Flames of Hate: Social Media and Hate Crime. *Political Economy - Development: Public Service Delivery EJournal*. http://dx.doi.org/10.2139/ssrn.3082972

Mundt, M. D., Ross, K. H., & Burnett, C. M. (2018). Scaling Social Movements Through Social Media: The Case of Black Lives Matter. *Social Media + Society*, *4*.

Naslund, J. A., Bondre, A. P., Torous, J. B., & Aschbrenner, K. A. (2020). Social Media and Mental Health: Benefits, Risks, and Opportunities for Research and Practice. *Journal of Technology in Behavioral Science*, 1–13.

Nobata, C., Tetreault, J. R., Thomas, A. O., Mehdad, Y., & Chang, Y. (2016). Abusive Language Detection in Online User Content. *Proceedings of the 25th International Conference on World Wide Web*.

Ousidhoum, N., Lin, Z., Zhang, H., Song, Y., & Yeung, D.-Y. (2019). Multilingual and Multi-Aspect Hate Speech Analysis. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4675–4684. https://doi.org/10.18653/v1/D19-1474

Pater, J. A., Kim, M. K., Mynatt, E. D., & Fiesler, C. (2016). Characterizations of Online Harassment: Comparing Policies Across Social Media Platforms. *Proceedings of the 19th International Conference on Supporting Group Work*, 369–374. https://doi.org/10.1145/2957276.2957297

Pavlopoulos, J., Malakasiotis, P., & Androutsopoulos, I. (2017). Deeper Attention to Abusive User Content Moderation. *EMNLP*.

Pavlopoulos, J., Sorensen, J., Laugier, L., & Androutsopoulos, I. (2021). SemEval-2021 Task 5: Toxic Spans Detection. *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, 59–69. https://doi.org/10.18653/v1/2021.semeval-1.6

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. https://doi.org/10.18653/v1/N18-1202

Poletto, F., Basile, V., Sanguinetti, M., Bosco, C., & Patti, V. (2020). Resources and benchmark corpora for hate speech detection: A systematic review. *Language Resources and Evaluation*. https://doi.org/10.1007/s10579-020-09502-8

Power, A., Keane, A., Nolan, B., & O'Neill, B. (2018). Detecting Discourse-Independent Negated Forms of Public Textual Cyberbullying. *Journal of Computer-Assisted Linguistic Research*, *2*, 1. https://doi.org/10.4995/jclr.2018.8917

Pruksachatkun, Y., Phang, J., Liu, H., Htut, P. M., Zhang, X., Pang, R. Y., Vania, C., Kann, K., & Bowman, S. R. (2020). Intermediate-Task Transfer Learning with Pretrained Language Models: When and Why Does It Work? *ACL*.

Radford, A., & Narasimhan, K. (2018). *Improving Language Understanding by Generative Pre-Training*.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. *EMNLP*.

Sap, M., Card, D., Gabriel, S., Choi, Y., & Smith, N. A. (2019). The Risk of Racial Bias in Hate Speech Detection. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1668–1678. https://doi.org/10.18653/v1/P19-1163

Sap, M., Gabriel, S., Qin, L., Jurafsky, D., Smith, N. A., & Choi, Y. (2020). Social Bias Frames: Reasoning about Social and Power Implications of Language. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5477–5490. https://doi.org/10.18653/v1/2020.acl-main.486

Saunders, M., Lewis, P., & Thornhill, A. (2016). *Research Methods for Business Students.* (7th ed.). Harlow: Pearson Education Limited.

Schmidt, A., & Wiegand, M. (2017). A Survey on Hate Speech Detection using Natural Language Processing. *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, 1–10. https://doi.org/10.18653/v1/W17-1101

Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. *Commun. ACM*, *63*(12), 54–63. https://doi.org/10.1145/3381831

Sellars, A. (2016). Defining Hate Speech. *Social Science Research Network.*

Siegel, A. A. (2020). Online Hate Speech. In N. Persily & J. A. E. Tucker (Eds.), *Social Media and Democracy: The State of the Field, Prospects for Reform* (pp. 56–88). Cambridge University Press.

Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., & Zhou, D. (2020). MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. *Proceedings of the 58th Annual*

*Meeting of the Association for Computational Linguistics*, 2158–2170. https://doi.org/10.18653/v1/2020.acl-main.195

Uhls, Y. T., Ellison, N. B., & Subrahmanyam, K. (2017). Benefits and Costs of Social Media in Adolescence. *Pediatrics*, *140*, S67–S70.

Ullmann, S., & Tomalin, M. (2020). Quarantining online hate speech: Technical and ethical perspectives. *Ethics and Information Technology*, *22*(1), 69–80. https://doi.org/10.1007/s10676-019-09516-z

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, \Lukasz, & Polosukhin, I. (2017). Attention is All You Need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *ArXiv*, *abs/1804.07461*.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-Attention with Linear Complexity. *ArXiv*, *abs/2006.04768*.

Warner, W., & Hirschberg, J. (2012). Detecting Hate Speech on the World Wide Web. *Proceedings of the Second Workshop on Language in Social Media*, 19–26.

Waseem, Z., & Hovy, D. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. *Proceedings of the NAACL Student Research Workshop*, 88–93. https://doi.org/10.18653/v1/N16-2013

Wirth, R. (2000). CRISP-DM: Towards a standard process model for data mining. *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, 29–39.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. *NeurIPS*.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). *Proceedings of the 13th International Workshop on Semantic Evaluation*, 75–86. https://doi.org/10.18653/v1/S19-2010

Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., Derczynski, L., Pitenis, Z., & Çöltekin, Ç. (2020). SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, 1425–1447. https://doi.org/10.18653/v1/2020.semeval-1.188

Zhang, T., Wu, F., Katiyar, A., Weinberger, K. Q., & Artzi, Y. (2021). Revisiting Few-sample BERT Fine-tuning. *ArXiv*, *abs/2006.05987*.

Zhao, J., & Gui, X. (2017). Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2017.2672677

Zimmerman, S., Kruschwitz, U., & Fox, C. (2018, May). Improving Hate Speech Detection with Deep Learning Ensembles. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. https://www.aclweb.org/anthology/L18-1404

# APPENDIX A.

The source code, data and results from this research can be found at the below public Github library.

https://github.com/D19124612/dissertation