****

**Automatically Detecting Hate Speech on Twitter Using BERT Word Embeddings**

**A dissertation submitted in partial fulfilment of**

**the requirements for the degree of**

**Bachelor of Science in Computer Science**

**in**

**The Queen's University of Belfast**

**by**

**Fionn C. McConville**

**29/4/2020**

*I plan to submit a paper based on this project to the Special Issue on Cyber-Social Health: Promoting Good and Countering Harm on Social Media of IEEE Internet Computing (*[*https://www.computer.org/digital-library/magazines/ic/call-for-papers-special-issue-on-cyber-social-health-promoting-good-and-countering-harm-on-social-media*](https://www.computer.org/digital-library/magazines/ic/call-for-papers-special-issue-on-cyber-social-health-promoting-good-and-countering-harm-on-social-media)*).*

Github Repository: [1]

CSC3002 – COMPUTER SCIENCE PROJECT

Dissertation Cover Sheet

Student Name: Fionn McConville Student Number: 40156103

Project Title: Automatically Detecting Hate Speech on Twitter Using BERT Word Embeddings

Supervisor: Dr Barry Devereux

Declaration of Academic Integrity

Before submitting your dissertation please check that the submission:

1. Has a full bibliography attached laid out according to the guidelines specified in the Student Project Handbook.
2. Contains full acknowledgement of all secondary sources used (paper-based and electronic)
3. Does not exceed the specified page limit
4. Is clearly presented and proof-read
5. Is submitted on, or before, the specified or agreed due date. Late submissions will only be accepted in exceptional circumstances or where a deferment has been granted in advance.

By submitting your dissertation, you declare that you have completed the tutorial on plagiarism at http://www.qub.ac.uk/cite2write/introduction5.html and are aware that it is an academic offence to plagiarise. You declare that the submission is your own original work. No part of it has been submitted for any other assignment and you have acknowledged all written and electronic sources used.

1. If selected as an exemplar, I agree to allow my dissertation to be used as a sample for future students.

Student’s signature: Fionn McConville Date of submission: 29/4/20

Contents

[1 Introduction 5](#_Toc39165552)

[1.1 Background of BERT 7](#_Toc39165553)

[1.2 System Requirements 8](#_Toc39165554)

[2 Related Work 10](#_Toc39165555)

[3 Data 11](#_Toc39165556)

[3.1 Fine-Tuning Datasets 11](#_Toc39165557)

[3.2 Limitations for Hate Speech Datasets 12](#_Toc39165558)

[3.3 Further Pre-Training Datasets 16](#_Toc39165559)

[4 Experiments 19](#_Toc39165560)

[4.1 Text Pre-Processing Stage 19](#_Toc39165561)

[4.2 Further Pre-Training Stage 22](#_Toc39165562)

[4.3 Fine-Tuning Stage 23](#_Toc39165563)

[4.3.1 Additions to the top of BERT’s Hidden Attention Layers 23](#_Toc39165564)

[4.3.2 Loss Functions to Correct Data Imbalance and Metric Imbalance 24](#_Toc39165565)

[4.3.3 L2 Regularization to Correct Possible Biases Towards the NSP task 25](#_Toc39165566)

[5 Evaluation upon the Training Set 25](#_Toc39165567)

[5.1 Experimental Setup 25](#_Toc39165568)

[5.1.1 Metrics 26](#_Toc39165569)

[5.1.2 Implementation Details 26](#_Toc39165570)

[5.1.3 Guaranteeing Reliable Results 27](#_Toc39165571)

[5.2 Text Pre-Processing 28](#_Toc39165572)

[5.3 Further Pre-Training BERT Embeddings 29](#_Toc39165573)

[5.3.1 Initial Learning Rate for Further Pre-Training BERT Embeddings 30](#_Toc39165574)

[5.3.2 Removal of In-Common Terms 30](#_Toc39165575)

[5.3.3 Comparison of Emoji Replacement Techniques 32](#_Toc39165576)

[5.3.4 Text Pre-Processing Approaches After Further Pre-Training 33](#_Toc39165577)

[5.4 Fine-Tuning 34](#_Toc39165578)

[5.4.1 Loss Functions to Correct Inferior Precision 34](#_Toc39165579)

[5.4.2 Techniques to Deal with Class Imbalance 35](#_Toc39165580)

[5.4.3 Additions on Top of BERT Hidden Attention Layers 36](#_Toc39165581)

[5.4.4 L2 Regularization to Normalise the Pooled [CLS] Embedding 36](#_Toc39165582)

[6 Results on Test Sets 37](#_Toc39165583)

[6.1 HatEval Dataset 37](#_Toc39165584)

[6.2 AnalyticsVidhya.com Dataset 39](#_Toc39165585)

[7 Error Analysis 40](#_Toc39165586)

[7.1 Non-identically Distributed Data 40](#_Toc39165587)

[7.2 False Positive Annotations 42](#_Toc39165588)

[7.3 False Negative Annotations 44](#_Toc39165589)

[8 Discussion 44](#_Toc39165590)

[9 References 46](#_Toc39165591)

**Abstract**

This paper explores the attempt to reliably and accurately classify hate speech online using word embeddings from Google’s BERT[[1]](#footnote-1) [1]. The text classification problem at hand is a binary choice*: Hate Speech or Not Hate Speech* - this sounds simple on paper, yet it has caused difficulty for people to define [2], which as a consequence extends to AI as it learns from humans [3]. This report demonstrates an in-depth analysis of the three main stages of the development of the classifier: the text pre-processing stage, the further pretraining stage and finally the fine-tuning stage, by attempting to find the optimum methodology at each step.

The pre-trained BERT-large model is utilized in this paper to transfer its learned knowledge of natural language onto the downstream task of classifying hate speech, with relatively simple architectural adaptation BERT can be fine-tuned to achieve near state-of-the-art performance in a range of NLP tasks. The end system performs competitively on a couple of datasets, it places 10th on the ongoing AnalyticsVidhya.com[[2]](#footnote-2) hackathon to combat hate speech out of ~1000 submissions and places 3rd for Task 5 of SemEval-2019: HatEval: Multilingual Detection of Hate Speech Against Immigrants and Women on Twitter for the hate speech detection sub-task out of 69 submissions.

# 1 Introduction

With the advent of social media, information, thoughts and opinions have never been shared as freely and globally. The potential of this is largely beneficial; it allows us as a society to organise public affairs, communicate with people around the globe and educate almost anyone regardless of wealth or social status. However, with the positive strides it allows us to make as a society, it would be remiss to ignore the negative consequences. As the ability to freely express ourselves escalates, so too does the exposure of harmful expression such as hate speech, which promotes illegal discrimination, causes distress to targeted individuals and groups and – in cases like the recent mosque shooting in Christchurch or the Finsbury park terror attack - incites violence and harm by contributing to these antagonists’ radicalisation [4] [5].

Regulators around the globe have scrambled for a solution to this metastasizing problem; some have sought to enforce real name policies [6], which will remove from users the anonymity that gives them the comfort to shamelessly spread detestable content that they would otherwise not utter in a public space. However, this has received some justifiable pushback from those who argue these measures harm the right to privacy, which as a consequence may even harm non-malicious actors [7].

Moreover, pursuing a strategy from a litigation point of view presents a lot of complications. As a result of the transnational reach of social media platforms such as Twitter and Facebook, it is difficult for governments to convince these companies to adhere to their understanding of what hate speech is. In the UK, it is a criminal offence to incite racial or religious hatred, and variations on this legislation exist in the majority of developed democracies including Australia, Germany, India, Sweden, and New Zealand [8]. However, it is deemed unconstitutional in the US to ban speech on these grounds because it conflicts with the first amendment [10]. Only when speech is considered to incite *violence* is legal action taken in the US.[[3]](#footnote-3)

Cross-jurisdictional co-operation on these issues among nations is therefore understandably either not forthcoming, or slow to work. “The longer the content stays available, the more damage it can inflict on the victims and empower the perpetrators. If you remove the content at an early stage, you can limit the exposure. This is just like cleaning litter; it doesn't stop people from littering but if you do not take care of the problem it just piles up and further exacerbates." [10]

Thus, it has become imperative that social media companies develop tools that quickly and automatically detect hate speech, using natural language processing (NLP), for commercial, regulatory and ethical reasons, as human detection and intervention is not realistically possible with the immense stream of user-generated content uploaded online every day[[4]](#footnote-4). Therefore, having an automatic NLP system to detect hateful posts will serve as a good first line of defence against hateful content online.

It is often difficult for even humans to distinguish a ground truth as to what constitutes as hate speech, as there are many differing opinions as to what exactly qualifies speech to be hateful [12]. Unlike other types of pernicious activity, such as spam or malware, hate speech is almost exclusively user-generated, rather than being created by bots [13]. This makes identifying this type of language less predictable and thus less identifiable.

Nevertheless, despite this uncertainty, a concise characterization of hate speech must be adopted for this study. The final NLP model that will be created will aim to identify hate speech based off the following definition:

Hate Speech is commonly defined as any communication that disparages a person or a group on the basis of some characteristic such as race, colour, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics [13].

## 1.1 Background of BERT

Transfer learning is a method in machine learning that heightens learning in a new task by transferring knowledge from a related task [15], this method of learning is especially applicable to natural language processing. Recently, vast advances have been made in NLP by using unsupervised language model pre-training to teach a model natural language understanding, then using the method of transfer learning to transfer this knowledge to a downstream task which relies upon interpreting language. Adapting this pre-trained model to a downstream objective is called fine-tuning and it is how BERT’s extensive knowledge of natural language understanding can be adapted to a vast array of NLP tasks.

BERT[[5]](#footnote-5) [1] is a contextualized word representation model that is based on a masked language model and pre-trained using bidirectional transformers [16]. Prior to BERT, previous language models were constrained to learning general language representations via a shallow concatenation of two unidirectional language models (i.e. left-to-right and right-to-left) in pre-training [16]. However, this unidirectional approach was sub-optimal for sentence level tasks as often understanding the context of a sentence must come from both directions, not just right-to-left or left-to-right or a superficial combination of both. BERT is a deeply bi-directional language model, as in pre-training it learns deeply bidirectional representations of language

BERT utilizes techniques from its predecessors in natural language modelling, combining the feature-based approach of ELMo [16] and the fine-tuning approach of OpenAI GPT [18]. BERT learns bidirectional context in language representations via two unsupervised learning tasks in pre-training:

1. By using a masked language model (MLM) that predicts randomly masked words in a sequence. To avoid each word being able to “see itself” and thus trivially predict the target word in a multi-layer context, 15% of the input words are [MASK] tokens.

2. Via a next sentence prediction task on the corpora on which it was pretrained, however as explained in detail in [Section 4.2](#FurtherPretrainstage), this task was not implemented when the classifier in this paper was further pre-trained.

Unsupervised learning via these tasks[[6]](#footnote-6) on a large corpus of words (BooksCorpus - 800M words) [18] and English Wikipedia (2,500M words) has enabled BERT to learn deep contextual representations for each word. These pre-trained word embeddings may serve well for a text classification task like hate speech, which is a deeply contextual subject itself.

Using the pre-trained language representations that BERT provides, excellent performance can be achieved through transfer learning on a corpus of supervised data. The default fine-tuning process for BERT is simple, first the model weights are initialized by using the pre-trained language model (BERT-Large) parameters, then from the final layer of BERT, the hidden representation of the [CLS] token[[7]](#footnote-7) is pooled by the pooling layer, a dropout of 0.1 is then used for regularization. Lastly, a fully connected feed-forward layer where the pooled output is multiplied by a weight matrix and bias is added, has its output fed to a softmax operation which normalises the output and gives probabilities that the sentence belongs to either class. In [Section 5.4](#finetuneeval) this paper explores whether this simple process can be improved upon by appending a more complex architecture to the top layer of BERT.

BERT embeddings were an attractive technique to adopt for this task as they often obtain state-of-the-art performance on most NLP tasks, while requiring minimal task-specific architectural modification. It was ground-breaking in the NLP field when it was released and recently many improved and more advanced pre-trained models base their design approach upon the groundwork which the authors of BERT established [20] [20].

## 1.2 System Requirements

Text pre-processing is crucial to NLP systems, this project attempts to investigate in detail the optimal pre-processing pipeline to implement when working with noisy user-generated tweets. Careful text pre-processing is undertaken so that hashtags and emojis – which often contain vital context to a tweet – could be interpreted by BERT. These methods of pre-processing evidently improved upon the basic text pre-processing that one would traditionally use in NLP applications as displayed in [Section 5.2](#textpreprocesseval).

BERT has already been pre-trained upon large text corpora using two tasks: word masking and next sentence prediction. As this model only classifies individual tweets, the next sentence prediction task in further pretraining is not necessary, so this project removes it. The current pre-trained BERT has not been exposed to informal grammar, emojis[[8]](#footnote-8) and slang which are commonplace online, as a solution, this classifier is further pre-trained on a large, meticulously collected tweet database which contains an abundance of such a vocabulary ([Section 3.3](#FurtherPretrainData)). Emojis are a domain specific vocabulary to online communication and appear frequently in tweets, in pre-processing they can either be represented as words (which already have pre-existing embeddings) or they can be represented as unique tokens ([Figure 9](#fig9)). This project investigates in depth which approach is superior, contrasting their effect on performance before further pre-training[[9]](#footnote-9) and after further pre-training using the respective emoji translation techniques on the fine-tuning data, wherein the unique token emojis will have a more appropriate weighted embedding having been exposed to BERT.

In addition, different approaches for further pre-training are deliberated, such as the choice of an initial learning rate or removing terms that overpopulate the corpora of tweets – (the motivation behind this is explained in [Section 5.3.2](#removeincommonterms). Experiments decisively demonstrate which methodology is more suitable in [Section 5.3](#furtherpretraineval).

The architectural choices that are built on top of BERT are also investigated, by attempting different fine-tuning models rather than just the simple addition of single feed-forward layer that the BERT classifier was built upon ([Section 5.4](#finetuneeval)). In particular, the addition of Bi-LTSM layers is experimented with to incorporate bi-directional long-term dependencies at the classification layer, which could prove useful in hate speech detection. Classifying hate speech online is inherently an imbalanced classification problem, software to detect hate speech must be developed with this in consideration. Accordingly, different loss functions and oversampling is experimented with to address this ([Section 5.4.2](#classimbalanceeval)).

State-of-the-art classification of hate speech was the ultimate end goal of this project. Whilst the designed system does not beat all comers, it does provide competitive results in popular hate speech detection shared tasks. The classifier, as well as any other hate speech detection software that is attempted by the open source community, is constrained significantly by the lack of hate speech data, both in sheer quantity and in reliability. This predicament is discussed comprehensively in [Section 3.1](#Finetunedata).

# 2 Related Work

Early studies to find the best model design for an automatic hate speech classifier applied the use of features like bag-of-words and an SVM classifier to detect racist content in web pages [21]. Following a similar approach, other early classifiers were developed using a combination of SVMs and word and character n-grams [22] [23] .

However, contemporary NLP models widely use pre-trained word embeddings as features, with deep learning architectures to fine tune them. [24] applied various word embeddings to tweets and used Long Short-Term Memory (LTSM) models to learn from them. This implementation resoundingly improved upon the previous state of the art word and character n-gram approaches by ~ 18 F1 points on a widely used benchmark dataset for hate speech. Another study [26], further demonstrated the supremacy of neural network based techniques by outperforming SVM baselines by 2-9% on several datasets by training a deep neural network combining CNNs with Gated-recurrent Units [27].

A large number of academic events, shared tasks and competitions have attempted to motivate the open source community to find an NLP based solution to hate speech and abusive language detection, in fact all the data intended for training in the fine-tuning stage of this project was sourced from events like these ([Section 3.1](#Finetunedata) explores these datasets in more detail). Several English language datasets have been authored [28], [28] and [30] with the express intent to contribute towards the research of this problem. Many more datasets have also been authored in different languages such as [30] at EVALITA 2018[[10]](#footnote-10) for investigating hate speech in Italian, [31] for the automatic misogyny identification shared task in Spanish[[11]](#footnote-11) and the GermEval Shared Task on the Identification of Offensive Language [32] which focussed on identifying offensive language in German.

The main dataset that is used for evaluation and testing throughout this paper is the HatEval dataset, from the Proceedings of the 13th International Workshop on Semantic Evaluation [34]. Many papers have used this set in their attempts to construct an automatic hate speech classifier, in total, 69 participant teams submitted predictions to the English hate speech detection task. The top performing system [34] used an SVM model with an RBF kernel to classify tweets using Google’s pre-trained Universal Encoder [35] sentence embeddings as features. The other best performing systems who provided a description of their system[[12]](#footnote-12) used neural network models as classifiers, with the third placed team *YNU DYX*, using stacked Bidirectional Gated Recurrent Units (BiGRUs) [27], with fastText word embeddings as input features [36], the output of BiGRU was then fed as input to the capsule network [38]. The fourth placed team went by the name of *alonzorz* e dahe

and used contextualised embeddings much like BERT. These embeddings were pre-trained upon 50 million unique tweets from the Twitter Firehose dataset and were fine-tuned using a novel type of CNN called a Multiple-Choice CNN.

# 3 Data

## 3.1 Fine-Tuning Datasets

The hate speech data for fine tuning the classifier were tweets; due to the abundance of available open source hate speech datasets online collected from twitter. Each dataset was labelled through majority labelling[[13]](#footnote-13) through crowdsourcing. These datasets are as follows:

**HatEval 2019** [34] – From SemEval-2019 shared task 5. The targets of hate speech in this dataset are either women or immigrants. The dataset was annotated from non-expert annotators from the crowdsourcing platform *Figure Eight* (F8), then the tweets were further reviewed by two more expert annotators. According to F8 there an intercoder-agreement score of 83% in the hate speech labelling of the dataset. 13K tweets overall (10K training, 3K testing), 4210 labelled HS

**OffensEval 2019** [38] - From SemEval-2019 shared task 6. This task was for categorizing offensive language on twitter rather than hate speech. There were three columns to describe the data: if the data was offensive, if it was a targeted insult/threat or not and if the target was a group, individual or other. Annotated by experienced annotators from the crowdsourcing platform Figure 8 (F8), the hypothesis was that if an entry were offensive, targeted and directed towards an individual/group then perhaps it could qualify as hate speech. 13.2K tweets overall, 3481 (potential) labelled HS

**ICVSM 2017** [28] - Hate speech dataset collected through filtering tweets that contained terms in the HateBase[[14]](#footnote-14)– a crowdsourced hate speech lexicon. To avoid false positives that occurred in prior work which considered all uses of particular terms as hate speech, annotators were instructed not to make their decisions based upon any words or phrases in particular, no matter how offensive, but on the overall tweet and the inferred context. The intercoder-agreement score provided by CF was 92%. 25K tweets overall tweets, 1430 labelled HS

**ICVSM 2018** [30] - Tweets in this set were labelled one of four categories: normal, abusive, hateful[[15]](#footnote-15) and spam. Annotated by users on Crowdflower, it is the largest by far of all datasets considered for the fine-tuning stage, it has 100K tweets, 4965 labelled HS.

**Waseem and Hovy 2016** [39] - Tweet ID datasets[[16]](#footnote-16). Datasets consisted of tweets labelled as either sexist, racist, both or neither. The tweets were collected by filtering twitter API to attain tweets that had one of 17 terms. The tweets were reviewed by the authors for HS, then further reviewed by experts[[17]](#footnote-17). Most of the racism tweets are targeting Muslim people and most of those that are considered sexist are criticising contestants on an Australian cooking show. [41]

**AnalyticsVidhya.com Practice Problem[[18]](#footnote-18)** – Very little information on the methodology followed for collection or annotation of tweet data. According to the website “For the sake of simplicity, we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it. So, the task is to classify racist or sexist tweets from other tweets.” The dataset is part of an ongoing competition so results can be benchmarked against other systems, also test data labels are unavailable prior to evaluation, so an honest evaluation of model performance can be realized.

## 3.2 Limitations for Hate Speech Datasets

The Nockleby hate speech definition (in the introduction section) and other quite closely related interpretations have been used to inform human annotators who label hate speech in datasets. Nevertheless, human annotators for several open source datasets online have struggled to label hate speech online appropriately [41]. Annotators for the datasets discussed in this paper – which are widely used in open source projects for hate speech detection - have often labelled contextually innocuous tweets containing terms previously used to disparage communities (such as the n-word and “queer”) as hate speech, despite them being reclaimed by those communities - although they remain offensive when used by outsiders. [41].

Also, terms that are simply used to identify communities based on race, gender and sexuality have been found to be disproportionally labelled as hate speech or toxic by automatic classifiers. Multiple studies have found that false positive labelling of hate speech are associated with the existence of these terms in text [43] [43]. False positive bias is likely caused by these terms being used in a pejorative sense online as well as the inherent biases of human annotators. [44] found biases in the Google Perspective API classifier[[19]](#footnote-19), trained on data from Wikipedia talk comments, gave high toxicity scores to harmless statements like “I am a gay man”.

This important context is often overlooked and hence the inherent biases that human annotators possess, are inherited by the models that are trained using these datasets, which ironically leads to discriminatory bias against the minorities that these hate speech detection systems were designed to protect [41]. Another related study found that AI models are one and a half times more likely to flag tweets as hateful or offensive when written by African Americans [45]. Even when these terms are not present in the tweet, the presence of an associated dialect with those terms, such as African American English (AAE) can be enough to trigger a false positive detection ([Figure 1](#Fig1)). Hate speech detection is 2.2x more likely when written in AAE according to the same paper.

Historically and politically contentious social media posts can often result in false positive detection, resulting in censorship ([Figure 3](#Fig3)). Even supposedly state of the art systems like Facebook’s proprietary hate speech filter can make this mistake – for example when Facebook’s filters flagged an excerpt from the Declaration of Independence as hate speech[[20]](#footnote-20) [46]. Instagram likewise faltered when its filters removed a post from an LGBT history account that posted the famous poem, I Want a President [47], which contains the opening line “I want a dyke for president” among other politically challenging statements. These errors in censorship may appear trivial, however they risk whitewashing history and restricting literature that is politically challenging and thought provoking [48].

Often tweets that were annotated as HS fell short of the definition that is guiding this project, with multiple examples of false positive annotations[[21]](#footnote-21). Occurrences of false positive HS in these datasets labelling seems to fall into 3 broad categories, which can also overlap with each other. Figures [1](#Fig1), [2](#Fig2) and [3](#Fig3) demonstrate this.

Figure 1

|  |  |
| --- | --- |
| ICVSM 2017: | RT @iamwilliewill: This what happens when you separate yo self from niggas who don’t eat they food cold. You FLOURISH…https://t.co/FzTIA |
| ICVSM 2018: | RT @ItsMeGrizz: Bad bitches don’t take days off https://t.co/eazGi8KnNh |

**There was an identity-based bias in the annotation i.e. the tweet was written in AAE and/or the tweet contained a term associated with derogatory intent.**

Figure 2

|  |  |
| --- | --- |
| ICVSM 2018: | I don’t give a fuck about NONE of y’all UGLY bitches at Riverdale lmfao get mad hoe |
| OffensEval 2019: | @USER you are a lying corrupt traitor!!! Nobody wants to hear anymore of your lies!! #DeepStateCorruption URL |

**The tweet was abusive towards an individual/group, but not targeting a group identity associated with hate speech.[[22]](#footnote-22)**

Figure 3

|  |  |
| --- | --- |
| AnalyticsVidhya.com: | Sikh temple vandalised in Calgary wso condemns act |
| Waseem & Hovy 2016: | RT @Dreamdefenders: Eric Holder from #ferguson: “I understand that mistrust. I am the Attorney General, but I am also a Black Man” https://t.co/eazGisia98 |

**The tweet was discussing or mentioning a politically contentious incident, not necessarily having hateful commentary on the incident.**

No dataset was perfect in its annotation, admittedly the inaptness of these examples is subjective opinion - the author of this paper is by no means an expert on annotating hate speech. This paper does not attempt to establish what dataset is right or what is wrong, instead the intent is to demonstrate that a deep analysis and methodology was followed for collecting hate speech data.

The discovery of false positive labelling as described in [Figure 1](#Fig1) concur with other papers on these datasets, [41] found that racial biases existed in the labelling of Waseem & Hovy 2016 [39], ICVSM 2018 [30] and ICVSM 2017 [28] – however the latter dataset was one of the few datasets tested that classified “white-aligned” tweets using racial epithets more than “black-aligned” tweets, (nevertheless there still existed a high racial bias towards tweets using an AAE dialect). Likewise, gender bias was found to be in the labelling of [39] by a related study [49].

Unfortunately, there is little definitive research on the other two categories of erroneous false positive labelling of hate speech (Figure [2](#Fig2) + [3](#Fig3)) in online HS datasets, likely because false positive labelling in these instances are subjective to the reader and are also difficult to prove through automatic NLP analysis. Furthermore, no detailed inspection of annotation on the AnalyticsVidhya or HatEval datasets could be found in literature.

To mitigate uncertainty in the annotation of the datasets, the main dataset that will be used for training and validation will be the HatEval dataset [34], as it contains an abundance of HS tweets and seems reliably labelled[[23]](#footnote-23) (intercoder-agreement score 83%). It may also be better to assess the performance of the final classifier by staying within datasets, rather than using a combined set, as each corpus has its own interpretation of hate speech. For example; data classified as not hate speech in the HatEval set may qualify as hate speech according to the annotators in [30]. [Figure 4](#Fig4) demonstrates this dichotomy.

Some evaluation will also be performed on the AnalyticsVidhya.com dataset, as it is an ongoing hackathon with thousands of contestants. As a result, the system designed in this paper can be benchmarked in real-time against other participant’s systems - which is an attractive prospect. Upon cursory inspection of the dataset, much of the content does not often seem to qualify as hate speech according to this paper’s strict definition, although there is a class imbalance problem the dataset has for “hate speech” labelled tweets, which is an interesting point of research for BERT[[24]](#footnote-24).

Figure 4

|  |  |
| --- | --- |
| ¬HS HatEval 2019 | Bitch you were supposed to be home 30mins ago you fat hoe |
| HS ICVSM 2018 | RT @ynaoivw: nah bitch I hate u https://t.co/fHX8y7esMH |

**Inconsistent Labelling Between Datasets - these two tweets roughly communicate a similar message. (Arguably the tweet in the HatEval dataset is more hateful).**

## 3.3 Further Pre-Training Datasets

Further pre-training BERT (explained in more detail in [Section 4.2](#FurtherPretrainstage)) on within-domain corpora has demonstrably improved performance in many fields with a specific vocabulary, such as in the biomedical field with BIOBERT [50] and for clinical text like Clinical BERT [51]. The everyday vernacular used on Twitter can be quite in contrast to the language used on a grammatically correct website like Wikipedia - upon which BERT is originally pre-trained - and thus in its own way, tweets can be thought of as having a more specific vocabulary. This is especially true for abusive language and hate speech, which often coincides with colloquial terminology that BERT is unlikely to have a contextual representation for in its pre-trained word and sentence embeddings. Hence further pre-training BERT on a large corpus of tweets may allow a much-improved understanding of language representation of hate speech on Twitter.

The additional data for further pre-training is not labelled for hate speech, nor is it required to be. The pre-training procedure involves self-supervised training on masked language modelling tasks, therefore the self-supervised learning in the further pre-training stage is not as dependent on the content of what it is being trained on, as much as the supervised data in the fine-tuning stage of the model development. As a result, the content of the datasets used in this stage are not heavily scrutinised like those in the fine-tuning stage, still they are targeted to be within domain on the target subject as further pre-training on a within-domain corpora[[25]](#footnote-25) has provided evident improvements upon the baseline BERT embeddings [52].

The intention when sourcing the pre-training datasets was to find tweet corpora that were largely user-generated and had a high likelihood of containing abusive or aggressive content with a possibility of including racial, gender or sexuality-based slurs. Tweets discussing politically contentious issues or events concerning minority populations were also targeted, as these controversies are often swelling with vitriol on both sides and are littered with the colloquial terminology and identity-based terms associated with hate speech [53] [54].

The datasets used to further pre-train the classifier are as follows:

**#UniteTheRight** [55] - The Unite the Right rally (also known as the Charlottesville rally) was a protest in Charlottesville, Virginia, United States from August 11–12, 2017, to oppose the removal of a statue of Robert E. Lee in Emancipation Park [56]. Protesters included white supremacists, white nationalists, neo-Confederates, neo-Nazis, and militias. This dataset contains 200,113 tweet ids collected with the #unitetheright hashtag.

**#Charlottesville** [57] - The same event as above but tweets were sourced from the hashtag #charlottesville. Many tweets in common are expected with the #unitetheright corpus but these will be removed in pre-processing. This dataset contains 200,000 tweets.

**#BLMKidnapping** [58] - These 136,990 tweet ids represent reaction to a Facebook Live video that was posted on January 3rd, 2017, showing four African American men violently attacking a white, mentally disabled man. After the video surfaced, the Twitter hashtag, #BLMkidnapping, was created and used to incorrectly attribute the violent attack to members of the Black Lives Matter movement. Police in Chicago, where the attack took place, have found no evidence the attack has any connection to the Black Lives Matter movement [59].

**Replies to Alexandria Ocasio Cortez Tweets** [60] **-** Whilst many tweets to Ms. Ocasio-Cortez may be glowing praise, it’s likely many tweets replying to her are abusive and perhaps sexist and racist, as America’s political climate has evidently grown increasingly toxic in recent years. 109,201 tweet IDs are in this dataset.

**#thechalkening** [61] **-** The Chalkening is a campaign launched by Donald Trump supporters on college campuses that involves writing pro-Trump messages in chalk on campus facilities. This mass, chalk-based, protest happened alongside an outpouring of media criticism of an incident at Emory University in March 2016 [62]. An Emory university administrator sent an email expressing support for students who claimed to feel threatened and unsafe by hate speech in the form of pro-Trump writings on the campus. This corpus contains 115,524 tweet IDs.

**#bill10** [63] **-** A list of 24,876 Twitter IDs for tweets containing the hashtag #bill10. Bill 10 in the Alberta legislature would have given public and Catholic school boards the right to refuse student requests to form gay-straight alliances in schools. Under intense public interest it was withdrawn by the Conservative government [64].

**#NotAllMen** [65] **-** Approximately 70,000 tweet IDs with the #notallmen hashtag. A Time magazine article on the subject states that "Not all men" was previously stated as an object of frustration, but in early 2014 it became usually used as an object of mockery [66]. Intended to counter generalizations about men's behaviour, some critics claim the phrase deflects conversations from uncomfortable topics, such as sexual assault [68].

**#YesAllWomen** [65] **-** This hashtag was popular in May 2014 and was created partly in response to the Twitter hashtag #NotAllMen. #YesAllWomen reflected a grassroots campaign in which women shared their personal stories about harassment and discrimination. The campaign attempted to raise awareness of sexism that women experience, often from people they know [68]. There are around 2.7M tweets in this database.

**Immigration and Travel Ban Tweet Ids** [69] **-** This dataset contains 16,875,766 tweet IDs, (yet this study only attempted to extract ~3,000,000 tweets) related to the immigration and travel ban executive order announced by the Trump Administration in January 2017 [71].

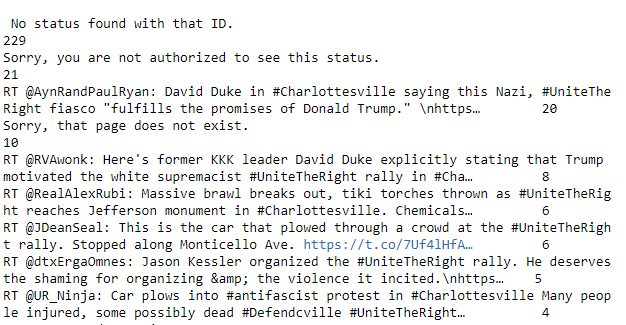
Also, the training sets for the hate speech datasets discussed in [Section 3.2](#Finetunedata) were used in further pre-training the BERT word embeddings[[26]](#footnote-26). The datasets listed above were tweet ID datasets[[27]](#footnote-27), so tweets were extracted from twitter via a twitter API tool[[28]](#footnote-28), were cleaned by removing duplicate tweets[[29]](#footnote-29) and removing null entries. The tweet text underwent identical text pre-processing to the optimal strategy found in [Section 5.2](#textpreprocesseval), to resemble the data that the classifier is fine-tuned on later. In total, 1,253,261 tweets were used to further pre-train BERT, more detail on the final, combined dataset for further pre-training is displayed in [Table 1](#table1)

Table 1 - Results of Cleaning the Further Pre-Training Datasets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DataSet | Total Tweets Before Data Cleaning | Total Tweets After Data Cleaning | % of Tweets Lost in Cleaning | % of Total Dataset |
| Combined Set | 671,180 | 274,026 | 59.4% | 21.9% |
| #theChalkening | 115,324 | 5,021 | 96.5% | 0.004% |
| #NotAllMen | 69,873 | 15,152 | 78.3% | 0.12% |
| #YesAllWomen | 2,705,985 | 286,315 | 89.4% | 22.3% |
| Immigration and Travel Ban | 3,000,000 | 570,140 | 81.0% | 45.6% |
| HS Datasets | 161,081 | 144,925 | 10.0% | 11.6% |
| **Total**  **Many of the tweet IDs in these datasets were either erroneously entered IDs, belonging to suspended or private accounts, or were retweets - Figure 6 reinforces this postulation. The “combined” set is a concatenation of the AOC replies, #unitetheright, #bill10, #BLMkidnapping and #charlottesville datasets. \*Approximately 45,000 tweets were removed after duplicates were** **dropped over the final, combined dataframe.** | **6,562,362** | **1,250,079\*** | **82.47%** | **100%** |

Due to the disorganised nature in how these tweet IDs are retrieved[[30]](#footnote-30), it is of no surprise that many of the tweets are either duplicates or unverified. Many accounts to whom tweets belong to in these datasets likely have been suspended in the wake of stricter hate speech regulations that twitter has imposed [71]. Also, the mass retweeting of a popular post could result in several duplicates ([Figure 5](#fig5)). These errors in the corpora were rectified before using this data for further pre-training

Figure 5: Content of a 1000 Tweet Sample of a Tweet ID Dataframe



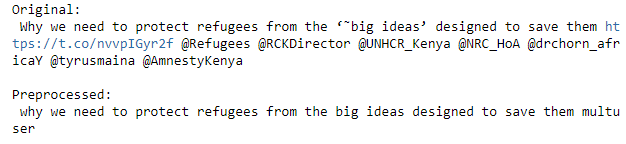
**Outcome of calling value\_counts() on a 1000 tweet sample of the combined[[31]](#footnote-31) database. 229 tweet IDs (22.9% of set) were belonging to no tweet, other tweets appear several times showing retweets were in the database frequently.**

# 4 Experiments

## 4.1 Text Pre-Processing Stage[[32]](#footnote-32)

Various text pre-processing strategies were experimented with to achieve optimal performance; relatively traditional techniques such as replacing mentions with common tags[[33]](#footnote-33), removing URLs, punctuation removal, removing retweet handles and converting to lowercase were attempted ([Figure 6](#fig6)). Removing stop-words and lemmatizing text were also experimented with, however these methods were deleterious to performance as proven in [Section 5.2](#textpreprocesseval)

Figure 6: Basic Pre-Processing

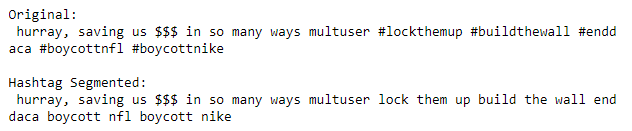


**The hyperlink and punctuation are removed after pre-processing. Also, the multiple tags – (which take up space in a sequence and thus detract importance from more informative tex)t – is replaced by the one instance of “multiuser” which has been added to the BERT vocab file so it can have its own word embedding.**

More complicated techniques such as emoji translation to text (Figures [8](#fig8) + [9](#fig9)) and hashtag segmentation ([Figure 7](#fig7)) were also successfully implemented with demonstratable improvements upon performance as shown later in [Section 5.2](#textpreprocesseval) and [Section 5.3.4](#textpreprocessafterfurtherpretrain). These techniques were sometimes vital to understanding context in a tweet as they would convey vital information in the hashtags e.g. #sendthemback was in many of the hate speech labelled tweets, yet would not be tokenized by BERT as the phrase “send them back”, which would have weighted representation in the pre-existing word and sentence embeddings in BERT, hence better recognised[[34]](#footnote-34). Likewise, emojis communicate sentiment, such as sarcasm or malintent, which can be vital to understanding the intent of a tweet.

Hashtag segmentation was achieved through the wordsegment[[35]](#footnote-35) library which is based on the unigram and bigram language algorithms covered in [72]. The system first obtains segmentation candidates that are scored using the n‐gram models, and then the best sequence of segmented words is selected using the Viterbi algorithm.

Figure 7: Hashtag Segmentation



Two separate emoji translation techniques were attempted, the first: emojiReplace\_v1 ([Figure 8](#fig8)), was a simple function which replaced emojis with their textual form using the emoji[[36]](#footnote-36) package. The second: emojiReplace\_v2 ([Figure 9](#fig9)) was a novel approach designed to take advantage of BERT’s vocab file[[37]](#footnote-37) - which had its first 1000 entries empty so users could modify the existing vocabulary by adding their own vocab.

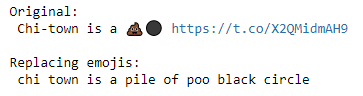
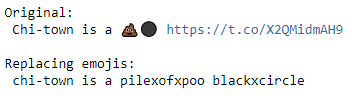


Figure 9: Tweet Before/After emojiReplace\_v2()

Figure 8: Tweet Before/After emojiReplace\_v1()

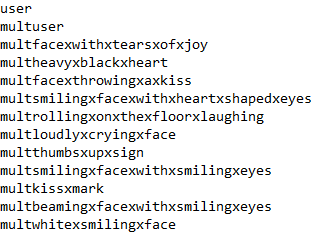
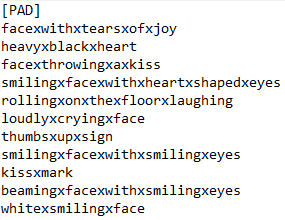
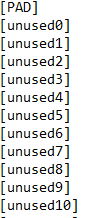
emojiReplace\_v2() replaced the most popular 500 emojis[[38]](#footnote-38) with unique tokens to represent them in singular and multiple format and then were added to the BERT vocab file ([Figure 10](#fig10)), which then would gain their own weighted word embeddings through further pre-training and fine-tuning as they would be mapped to a word ID ([Figure 11](#fig11)) which has its own weighted vector. A comparison of these two methods before and after further pre-training is analysed later in Sections [5.2](#textpreprocesseval) and [5.3.3](#emojieval) respectively.

Figure 11 New Tokens Mapped Successfully to Unique Word IDs



**Emoji token multxloudlyxcryingxface = 505**

Figure 10 – Modified vocab.txt file:



**Replace unused tokens with unique emoji token identifiers which represent individual emojis and consecutive emojis – which appear often on twitter. Likewise, the common tags of “user” and “multiuser” are added to the vocabulary**

It would be remiss to not mention that BERT’s vocab file controls for terms that could be out of vocabulary by breaking unrecognised words into wordpiece embeddings [73], i.e. "calling" -> ["call", "##ing"]. The embeddings for these workpieces already combine into a quite adequate representation, in a workpiece vocabulary because words are broken into pieces, there are no out-of-vocabulary words, but it was the goal of this paper to see if adding vocabulary specific tokens like emojis, which appear often, could enhance performance.

## 4.2 Further Pre-Training Stage

As mentioned in [Section 3.3](#FurtherPretrainData), further pre-training BERT has exhibited a large performance increase in previous studies [52], especially when classification is on a target task which contains a domain specific vocabulary [50] [51]. Unlike [52] who also attempted to optimise BERT for text classification, this study chose to effectively turn off the next sentence prediction (NSP) task in further pre-training[[39]](#footnote-39) as NSP is irrelevant to classifying individual tweets. Furthermore, the authors of an improved version of BERT - ROBERTA [20] showed that removing the NSP loss matches or slightly improves downstream task performance.

Tweet datasets for further pre-training were all were sourced via a Twitter API search through in-common terms such as #YesAllWomen or #Charlottesville. The concern was that by further pre-training BERT on these datasets with repeated terms, its masked language predictions may learn to over-predict these terms and thus harm natural language understanding. On the other hand, removing these terms may remove a lot of semantic meaning from these tweets which could also harm the natural language understanding of the model. Thus, in-common terms were removed from the pretraining data to see if there was a performance improvement or deterioration when further-pretraining; both approaches were tested in [Section 5.3.2](#removeincommonterms).

Experiments were also carried out in this stage to determine what the best initial learning rate to further pre-train was. The BERT repository suggested an initial rate of 2e-5 - however a learning rate of 5e-5 was also tested to see if there were improvements ([Section 5.3.1](#learnratefurtherpretrain)), despite concerns that too aggressive an initial learning rate could result in catastrophic forgetting[[40]](#footnote-40) [74]. Furthermore, a comparison was made in further pre-training and fine-tuning on the two separate emoji translation functions ([Section 5.3.3](#emojieval)) to determine if there was much of a performance gain after exposing the pre-training stage to text manipulated by these methods of emoji translation, specifically the translating emojis to tokens function, which converted emojis to unique tokens that BERT embeddings had not encountered before.

## 4.3 Fine-Tuning Stage

### 4.3.1 Additions to the top of BERT’s Hidden Attention Layers

Hoping to build upon the research of [76] whom didn’t find a successful way to append a bidirectional LTSM or extra hidden layers to the final attention layer of the BERT model, this paper attempted to optimise the fine-tuning layer by adding a more complex neural network strategy to better learn from BERT word embeddings. The approach of using an LTSM is an attractive prospect, as unlike feed-forward neural networks (which is the default classifier when fine-tuning BERT), recurrent neural networks like LSTMs can use their internal memory to process arbitrary sequences of inputs. Hence, LSTMs can be utilized to capture long range dependencies in tweets, which could prove useful in hate speech detection [24].

As the authors of [76] postulated when reflecting on their failure to successfully use a bi-LTSM[[41]](#footnote-41) with BERT, a bi-LTSM layer would have to be applied whilst using techniques to avoid overfitting, via careful hyperparameter selection. Hence, when extra hidden layers were added to the final layers of BERT in fine tuning in this study[[42]](#footnote-42), they had an increased dropout of 0.2 and 0.3 and early stopping was implemented to determine the correct number of steps for each model setup. [Section 5.4.3](#addingBERTlayerseval) showcases the results of these experiments

### 4.3.2 Loss Functions to Correct Data Imbalance and Metric Imbalance

Cost-sensitive loss functions were added to the classification layer of BERT to see if there were any changes in performance, the default loss function was a standard cross entropy calculation. Having a loss function to reflect imbalance may be beneficial – especially when the classification of hate speech on twitter online is often an inherently imbalanced classification problem [30]. Appropriate adjustment for imbalance was especially essential when dealing with the AnalyticsVidhya dataset, which had a class ratio of around 13:1 in the training set.

Accordingly, focal loss [76] was attempted as a solution; this loss function was designed by Facebook AI to deal with class imbalance by down-weighting the loss assigned to well classified examples. The goal in doing this was to prevent a vast number of easy true negatives overwhelming the classification layer in training. By putting more of an emphasis on data that is hard to classify, the focal loss function automatically assigns minority classes a better weight during training, thus improving recall and overall F Score. Focal loss was initially developed for dense object detection; however, it has demonstrated benefits in NLP applications [78], so it was used in this study.

A weighted[[43]](#footnote-43) cross-entropy loss function, was also experimented with to address this problem, the weights for positive and negative samples were adjusted to reflect the class imbalance to upweight less frequently occurring hate speech. Finally, the simple method of randomly oversampling the minority class was attempted to correct class imbalance by synthetically creating more minority class instances to match the majority class. The results from these experiments are in [Section 5.4.2](#classimbalanceeval)

These loss functions, whilst promising effectiveness in dealing with imbalanced data, can also help remedy an imbalance in metrics. Throughout cross-validated evaluations upon the HatEval dataset, the recall metric exceeded that of the precision metric. Whilst a loss function like focal loss has been designed in such a way to automatically address a problem like this, the weighted loss function had to be tuned for it. Results from these experiments are in [Section 5.4.1](#lossfunctionsforprecision)

### 4.3.3 L2 Regularization to Correct Possible Biases Towards the NSP task

A study [79] asserts that there is a mismatch in training objectives in the original pre-training of BERT embeddings, as these embeddings were pre-trained on masked language modelling and next sentence prediction. The latter task, as discussed before, is irrelevant to classifying individual tweets and so there is bias in the BERT embeddings towards the irrelevant NSP objective. This biased embedding can bring difficulties to the optimization process during fine tuning as the gradients of the [CLS] embedding may explode and result in a decline in performance. As a solution, they propose L2 normalisation of the [CLS] embedding in fine-tuning, which they demonstrate can improve BERT for text classification tasks. [Section 5.4.4](#L2regeval) tests this assertion upon the HatEval dataset

# 5 Evaluation upon the Training Set

## 5.1 Experimental Setup

All the evaluation in this section is performed upon the HatEval dataset mentioned in previous sections, unless specified otherwise. Steps for each evaluation was set at 850 steps – which was decided to be the optimal number of steps by early stopping. This was changed in fine-tuning where alterations to model architectures meant that 850 steps were no longer optimal for that configuration, e.g. for a Bi-LTSM there would be less steps because by 850 steps the model would overfit. The basic configuration of the model also had a dropout of 0.1 and a loss function of categorical cross-entropy for most experiments, however these hyperparameters were explicitly altered when exploring different methods in the fine-tuning stage ([Section 5.4](#finetunestage)). The rest of the hyperparameters in [Table 2](#table2) stayed constant throughout.

Table 2: Constant Hyperparameters in Fine-Tuning

|  |  |
| --- | --- |
| Batch size | 32 |
| Fine-Tuning Learning rate | 2e-5 |
| Optimizer | Adam |
| L2 weight decay | 0.01 |

### 

### 5.1.1 Metrics

The results of the models were evaluated based on four metrics: the accuracy, F1 score, precision and recall. The metrics can be computed as follows:

The F1 Score is the harmonic mean between the precision and recall and is metric by which each model implementation will be judged. The model design at each step will be determined by the method at each stage which achieves the highest F1 score.

### 5.1.2 Implementation Details

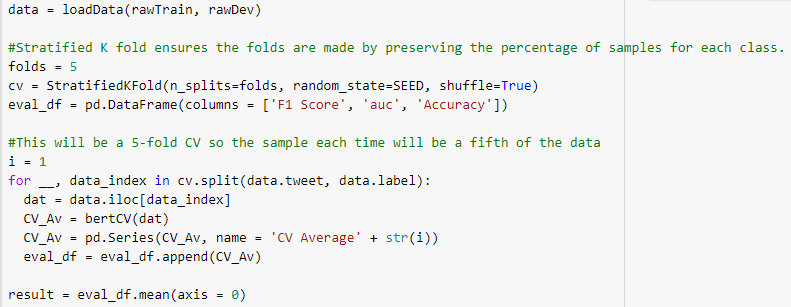
Python will be used to complete this project using Jupyter notebooks to collect tweets and visualisations and using Google Colab to run the memory intensive BERT models. Google Colab is essential to this project as it provides access to a Cloud TPU (Tensor Processing Unit) which allows for distributed training of the model shared among 8 cores of the TPU. Without this, training such a large model like BERT-Large (24-layer, 1024-hidden, 16-heads, 340M parameters) would be impossible as training upon a CPU or even GPUs result in an out-of-memory error (OOM). Visualisations throughout this project are completed using the Seaborn library, sklearn is used to collect metrics and the Tweepy library is used to collect tweets via Twitter API

The original BERT model is coded in Tensorflow1.x and uses the Tensorflow estimator API in the original repo for training, evaluation and prediction. These design choices are adopted in this project also, especially as the estimator API works well with TPUs. Also as a consequence, neural network additions to BERT’s architecture such as fine-tuning BERT embeddings using an LTSM, are coded in Tensorflow 1.x. Due to the virtual nature of Google Colab, data had to be stored in a virtual location; GCS buckets were used for this purpose. Customized functionality of the BERT library was required throughout this project, whether it was a choice of fine-tuning strategy, further pre-training without next sentence prediction or returning a more detailed set of metrics than the default implementation of BERT was designed to do[[44]](#footnote-44), hence the BERT repo was forked to this project’s Gitlab repository [1] and was edited and utilized there.

### 5.1.3 Guaranteeing Reliable Results

Many studies have concurred that non-determinism is prevalent in the Tensorflow framework and thus results are extremely difficult to reproduce run-to-run. Deep learning frameworks have components which have a stochastic nature, such as gradient descending algorithm in the training of a model, which causes variance in performance metrics [80]. The non-determinism of Tensorflow 1.14 proved to be a dilemma when attempting to collect reliable results in experiments. Normally k-fold cross-validation would remedy this, but even then, the variance in metrics run-to-run was greater than the performance improvements made by parameter tuning throughout this study. Therefore, the results collected in experiments were ran through a 5-fold nested cross-validation loop - the code which implements this is in [Figure 12](#fig12).

Figure 12: 5-fold Cross-Validation Within a 5-fold Cross Validation:



**Each fold was stratified, meaning that each subset of the data had an almost equal distribution of hate speech and non-hate speech.**

The outer loop partitions the data into 5 stratified subsets and each subset has 5-fold cross-validated metrics calculated from it which are then averaged again, resulting in performance estimates that will be much more dependable. A study which explored this validation strategy called it “J-K fold CV” and empirically demonstrated significant decreases in variability run-to-run [81]. This extensive process assured reliable metrics were collected for every implementation of the model attempted.

Within a cross-validated loop, one training fold could seldom have the potential to return a model which predicts one class, thus returning a poor set of metrics and resulting in an unfair evaluation of model performance, often through no fault of parameter choice. Accordingly, when this occurred, the metrics for that fold were ignored and were not involved in the calculation of the overall cross-validated evaluation average.

An enormous amount of tweet IDs were processed in the collection of data for the further pre-training stage of the model ([Section 3.3](#FurtherPretrainData)). As these tweet datasets were collected in an unsupervised manner, they were populated with many errors and duplicate tweets. These were appropriately disposed before these tweets were used. Also, to ensure data was collected, when processing the colossal tweet datasets that contained more than 1,000,000 tweets a checkpointing practice was adopted wherein every 500,000 iterations a csv containing all the tweets was saved to virtual storage on GCS. The test data was not used in any further pre-training

## 5.2 Text Pre-Processing

Each text pre-processing strategy was initially experimented with before further pre-training BERT, so that an optimal pre-processing pipeline could be identified and implemented on the further pretraining corpus. Research found that Hashtag Segmentation + ReplaceEmojisWithWords + Punctuation Removal was the best method of pre-processing before further pre-training, albeit by a marginal amount compared to the other methods ([Figure 13](#fig13)).

Lemmatization and removing stop-words were quite harmful approaches of pre-processing, with both additions making performance worse than the baseline (-0.24% and -4.77% reduction in F-Score respectively). This is possibly due to the noisy nature of tweet text, with sentences littered with abbreviations and irregular forms - which these methods struggle to deal with [82]. Also, the general text the pre-trained BERT model is trained on is not pre-processed in this way.

Hashtag segmentation and each emoji replacement function, (individually and combined), improve upon the baseline which was encouraging. Hashtag segmentation may have had the greatest enhancement upon performance because more crucial sentiment is communicated in hashtags than emojis when the subject is hate speech. On its own, hashtag segmentation improved upon the baseline F score by +1.96% and when coupled with removing punctuation and replacing emojis with words, the improvement was +3.34%.

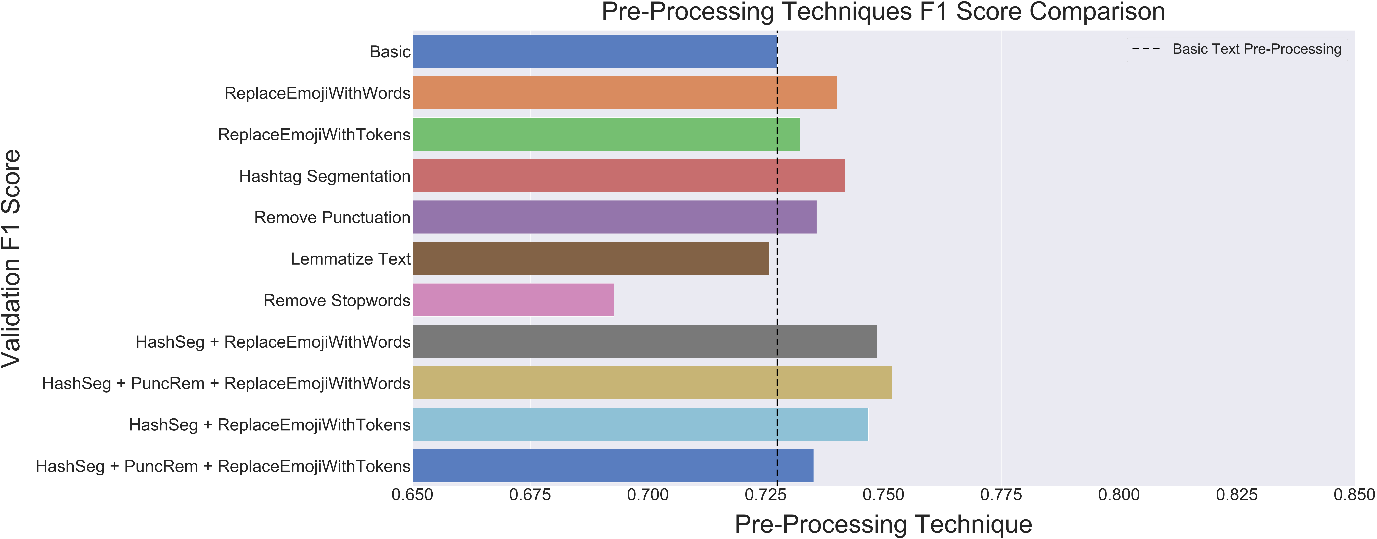


Figure 13: Text Pre-processing Performance TrhoEvaluation on Training Set

**Baseline (Basic) text pre-processing – (which consists of converting mentions to common tags, removing excess Unicode and URLs, converting text to lowercase and removing retweet handles) - was coupled with each method. Like every metric collected in this study, the F Scores were acquired through nested cross-evaluation**

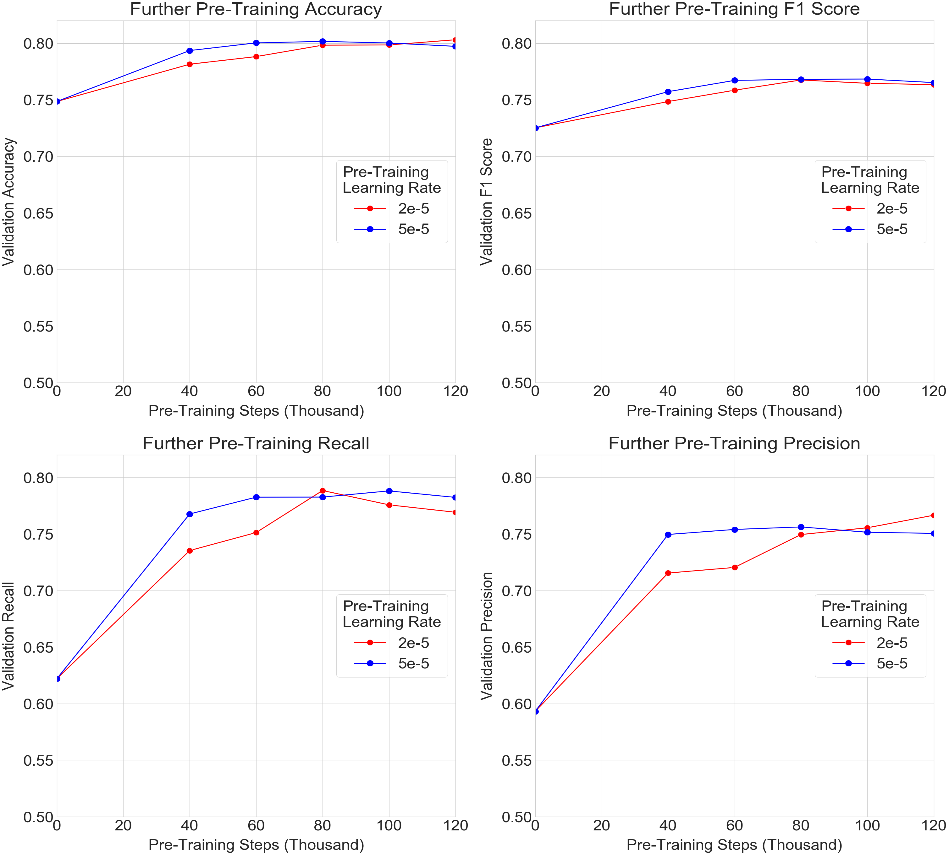
## 5.3 Further Pre-Training BERT Embeddings

As shown in [Table 1](#table1), around 1.25M within-domain tweets were attained for the further pre-training stage. For each experiment in this section, further pre-training was executed for 120K steps on this data; the amount of steps was decided upon because runtimes were limited on Google Colab TPUs[[45]](#footnote-45) and because [52] suggested that 120K was the best amount of steps to stop further pre-training BERT-large on[[46]](#footnote-46). Checkpoints were made after 20,000 steps each time in further pre-training, so that performance could be compared across each checkpointed model. The metrics gathered in this section are from the nested cross-validation metrics on the downstream hate speech classification task for the HatEval data, after conducting further pre-training with the language modelling objective. Further pre-training after 80,000 steps was consistently found to have the best F1 score across all experiments as shown in the figures in this section.

### 5.3.1 Initial Learning Rate for Further Pre-Training BERT Embeddings

As discussed in [Section 4.2](#FurtherPretrainstage) experimenting with different initial learning rates in further pre-training was a point of curiosity. Analysis shows that a learning rate of 5e-5 was marginally the most favourable initial learning rate to use, assuaging fears that too aggressive an initial learning rate might result in catastrophic forgetting ([Figure 14](#fig14)).

Figure 14: Comparison of Different Initial Learning Rates for Further Pre-TrainingBERT



**The learning rate of 5e-5 slightly outperforms 2e-5 by most metrics**

The difference in performance is very slight, however it is a consistent difference. More steps in further pre-training could have perhaps been carried out to see if the trajectory of performance, relative to more pre-training steps, resulted in a consistent plateau for the F-Score and accuracy metrics. It seems likely though that it would have remained this way for both initial learning rates, as when recall increases for one precision, the deteriorates for another after 80K steps.

### 5.3.2 Removal of In-Common Terms

The hypothesis that the inclusion of in-common terms like hashtags - that were used to source each dataset (e.g. #charlottesville or #thechalkening), would harm the learning of the word masking task in further pre-training, due to an over-prediction of these terms, was proven to be false. Removal of these in-common terms resulted in an inferior further pre-trained model ([Figure 15](#fig15)).

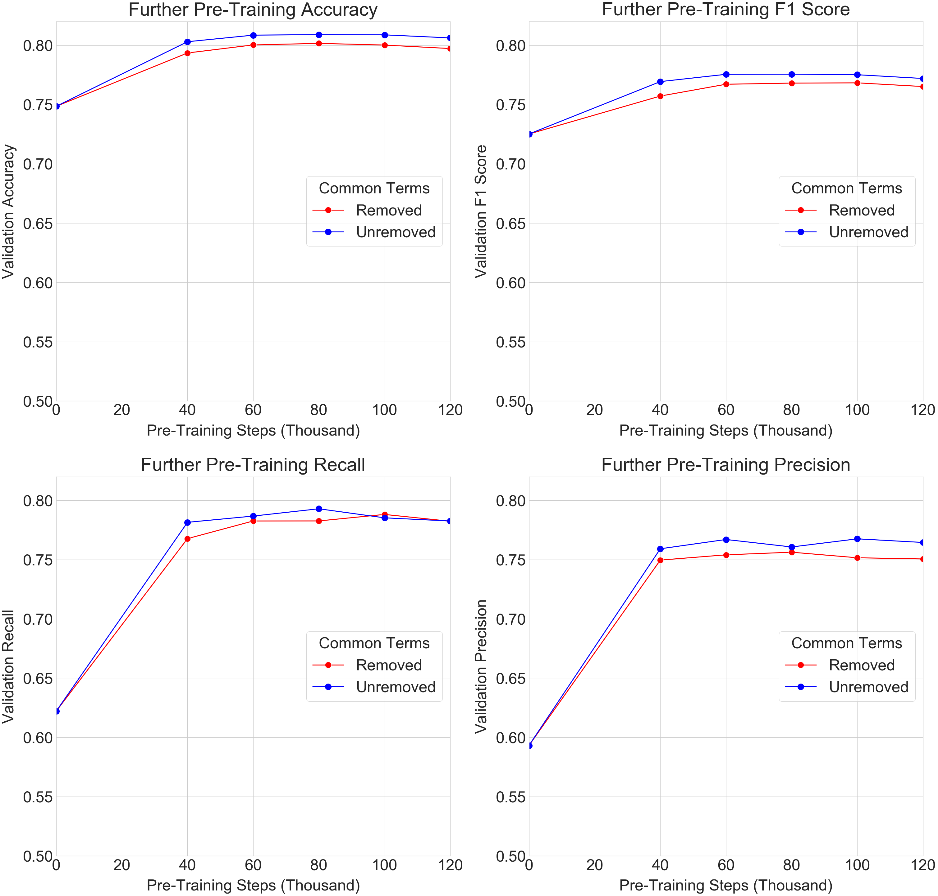


Figure 15: Removed Vs Unremoved Common Terms

**Not removing common terms is consistently better by most metrics apart from recall where it is relatively level**

In hindsight, removing these in-common terms was a rather brusque way to deal with the overpopulation of them in tweets. Often the result of such pre-processing lead to tweets like [Figure 16](#fig16), where much of the semantic meaning of the tweet is completely lost. It is likely that this loss in semantic meaning made the tweets affected in this way unsuitable for further pre-training the model as it would have been detrimental to natural language understanding.

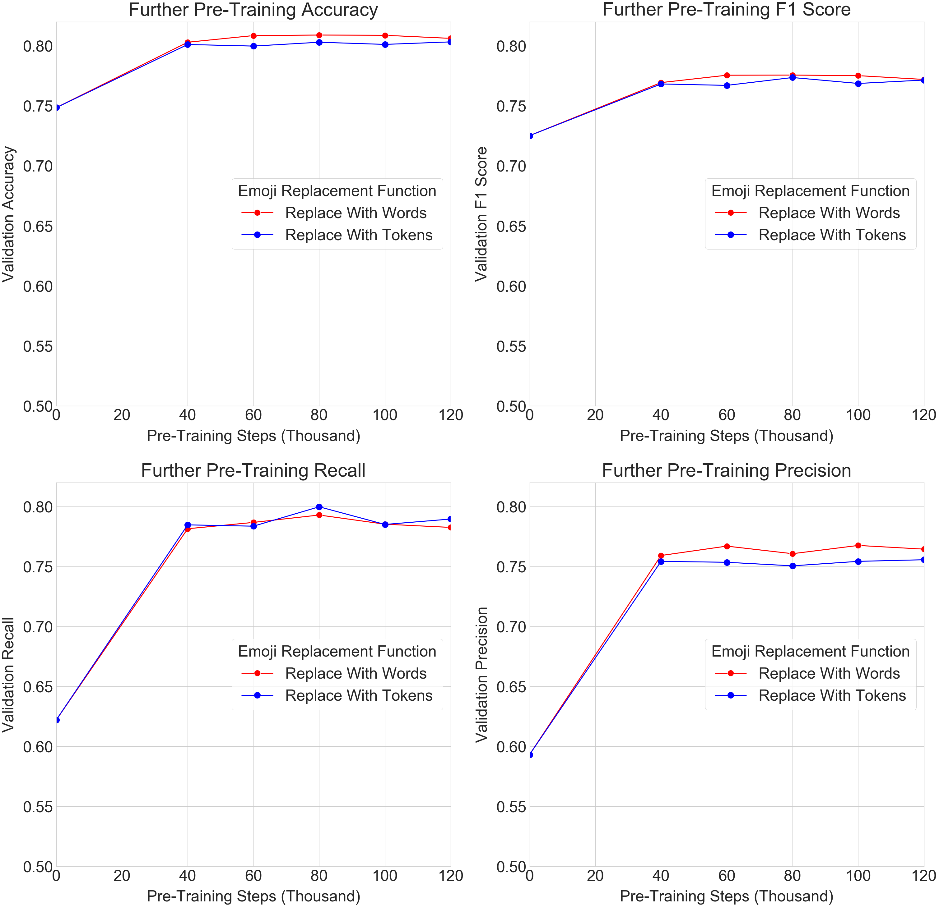
Figure 16: Before/After Removing Common Terms

|  |  |
| --- | --- |
| Before: | RT @Eidelonn: #notallmen are rapists but ANY MAN could be, from anywhere and at any time, so #YesAllWomen live in fear |
| After: | RT @Eidelonn: are rapists but ANY MAN could be, from anywhere and at any time, so live in fear |

**Common terms removed are #YesAllWomen and #NotAllMen**

### 5.3.3 Comparison of Emoji Replacement Techniques

Before further pre-training, replacing emojis with words ([Figure 8](#fig8)) appeared to be the optimal way to pre-process emojis, however it was stipulated this this may have been because these words had better pre-existing word embeddings in BERT than the unique tokens that were hard coded into the vocab file ([Figure 10](#fig10)). These unique tokens would have had randomly initialised weightings and would have only been tuned when fine-tuning the model on the data and so they didn’t have the advantage of undergoing an extensive pre-training procedure that words that pre-existed in BERT’s vocabulary had. Therefore, extensive experiments were carried out to see if further pre-training the model on data pre-processed with emojis being replaced by these unique tokens could perhaps enable this method of replacing emojis to surpass replacing emojis with words. Both methods were coupled with hashtag segmentation in the pre-processing of the data for this analysis and used identical initial learning rates for further pre-training and fine-tuning ([Figure 17](#fig17)).

Figure 17: Comparing Different Emoji Replacement Methods

**Replacing emojis with words was consistently a better method of pre-processing at all stages of further pre-training.**

Even after further pre-training, the strategy of replacing emojis with words remained consistently a better approach than the novel method of replacing them with tokens, apart from the recall metric where it matched and even sometimes bested replacing emojis with words. This inferiority may have been caused by tweets which had emojis that were not in the top 500 emojis online and so may have been out of vocabulary as only the top 500 emojis had unique token representation. These uncommon emojis would have had adequate representation in word form but not in unique token form ([Figure 18](#fig18)).

Figure 18: Comparison of Emoji Replacement Techniques Dealing with Rare Emojis



**Replacing emojis with tokens:**

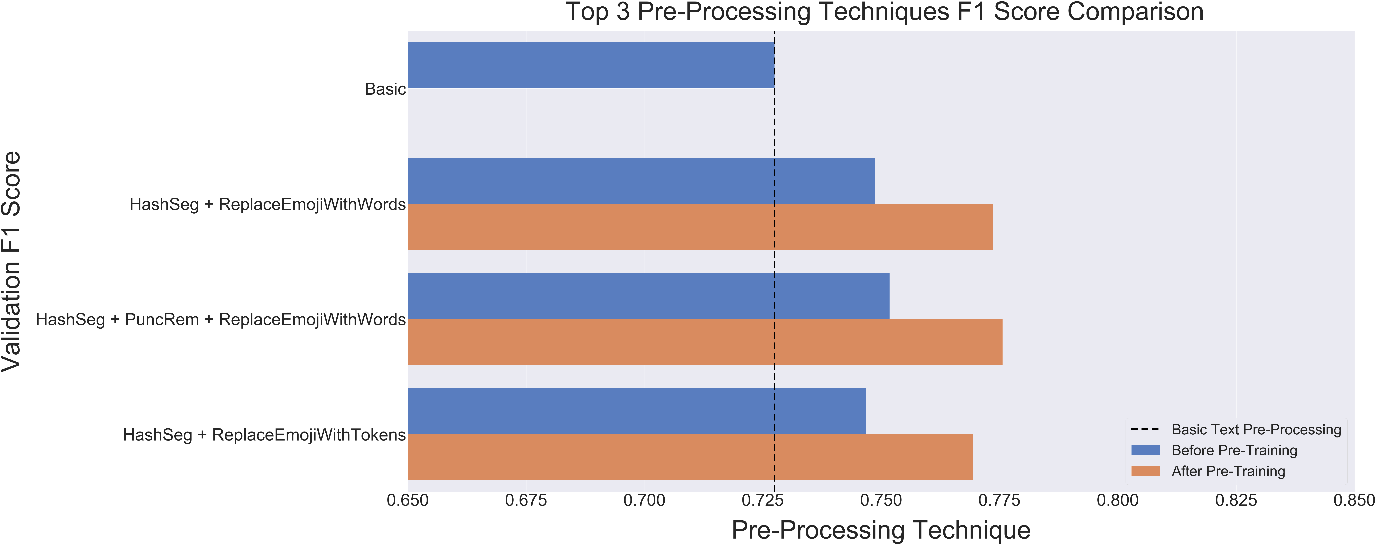
**Replacing emojis with words:**



**Replacing emojis with tokens in cases like these translates into a less coherent sentence**

### 5.3.4 Text Pre-Processing Approaches After Further Pre-Training

Figure 19: F Score Before and After Further Pre-Training for Top 3 Pre-Processing Techniques

 **These pre-processing pipelines were each used on pre-training and fine-tuning data.**

The optimal text pre-processing strategy remains a combination of hashtag segmentation, punctuation removal and replacing emojis with words as [Figure 19](#fig19) displays. Even when allowing BERT embeddings to become accustomed to the new tokens to represent emojis it still does not benefit them. The replacing emojis with words technique not only maintains its supremacy, but also it increases in performance by a wider margin ([Table 3](#table3)).

Table 3: F Score Before and After Further Pre-Training for Top 3 Pre-Processing Techniques

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-Processing Strategy | Before Further Pre-Train F-Score | After Further Pre-Train F-Score | Improvement After Further-Pretraining (%) | Improvement upon Baseline (%) |
| HashSeg + ReplaceEmojiWithWords | 0.749 | 0.774 | **+3.33%** | +6.34% |
| HashSeg + PuncRem + ReplaceEmojiWithWords | **0.752** | **0.776** | +3.17% | **+6.41%** |
| HashSeg + ReplaceEmojiWithTokens | 0.747 | 0.769 | +3.03% | +5.45% |

**Baseline is the default pre-processing method detailed in section --- with no pre-processing**

## 5.4 Fine-Tuning

Each result in the fine-tuning stage was run on the optimal text pre-processing and further-pretraining pipelines discovered in Sections [5.2](#textpreprocesseval) and [5.3](#furtherpretraineval) respectively, unless explicitly said otherwise. Each metric was collected through the nested cross-evaluation procedure detailed in [Figure 7](#fig7).

### 5.4.1 Loss Functions to Correct Inferior Precision

Loss functions can be used in training a machine learning classifier to deal with metric imbalance by allowing one to trade off recall and precision by up or down-weighting the cost of a positive error relative to a negative error [83]. Initial findings (using the default loss function in BERT fine-tuning whilst training on the HatEval dataset) were that the recall outmatched the precision of the classifier, despite the negative class being a majority class ([Table 4](#table4)). As a result, weightings were adjusted 2:1 for ¬HS:HS labelled tweets so that precision could increase, although this was expected to come at the expense of the recall of the classifier. Likewise, as the focal loss down-weights examples in training that are easy to classify, it was expected that the addition of focal loss to the classification layer might benefit the classifier’s precision, which it struggled with relative to recall.

Table 4: Different Strategies for Resolving Lower Precision - HatEval Set

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Loss Function | Accuracy | F1 Score | Precision | Recall | Steps |
| Default Cross-Entropy | 0.803 | 0.770 | 0.757 | **0.786** | **850** |
| **Focal Loss** | **0.806** | **0.773** | 0.764 | 0.784 | 700 |
| Weighted Loss | 0.806 | 0.769 | **0.772** | 0.768 | 700 |

**The weighting in the weighted loss function was 2:1 neg:pos and gamma was set to 2 for focal loss**

Focal loss seemed to be the best addition as it increases the F score metric the most, striking a good compromise between recall and precision. Unsurprisingly, the addition of focal loss and weighted loss have resulted in the down-weighting of positive examples, which makes the recall lower as a result. To ensure a fair comparison before cross-validated evaluation, early stopping was implemented on the training set, with a holdout development set to evaluate to decide the optimal number of steps for each different configuration.

### 5.4.2 Techniques to Deal with Class Imbalance

For the AnalyticsVidhya.com dataset, instead of the weightings in the loss functions being used to rectify an imbalance of metrics, it was used as a remedy for class imbalance, which had a ratio of 13:1 (¬HS:HS) in the training set. Far fewer steps were then necessary for the classifier to learn for the underrepresented HS class as it was significantly upweighted, this was also the case for the addition of focal loss which operates in a likewise fashion. Oversampling was also attempted as a solution for imbalance in the training set and it proved the most effective method in terms of F1 Score, although it came at the cost of recall ([Table 5](#table5)).

Table 5: Different Strategies for Resolving Class Imbalance - AnalyticsVidhya.com Set

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Accuracy | F1 Score | Precision | Recall | Steps |
| Default Cross-Entropy | 0.969 | 0.758 | 0.822 | 0.703 | 5200 |
| Focal Loss | 0.972 | 0.780 | 0.705 | **0.874** | 2400 |
| Weighted Loss | 0.971 | 0.785 | 0.804 | 0.768 | 2100 |
| **Random Oversampling** | **0.972** | **0.793** | **0.833** | 0.757 | 2200 |

**For weighted loss, class weighting was used to set the loss weights at 13:1 pos:neg upweighting the minority HS samples and for focal loss, gamma was set to 2. Oversampling was random and created more of the minority HS labelled samples to match the size of the majority class**

These techniques have resulted in a marked improvement upon the default configuration of BERT. The evaluation upon the imbalanced training set suggests that BERT’s default method of fine-tuning is relatively not very well equipped to deal with class imbalanced datasets and so should be tuned appropriately.

### 5.4.3 Additions on Top of BERT Hidden Attention Layers

To avoid the pitfalls of overfitting, the dropout of neurons in the added layers was increased to 0.2 and 0.3 to see if there was any improvement. Furthermore, less steps were used in fine-tuning the model, as these fine-tuning architectures often overfitted much quicker to the data than the default method of using only one layer in fine-tuning. Also, the number of steps per configuration was decided via early stopping, the hidden layers was fixed at 256 neurons for each model and ReLU was used as the activation function within the added hidden layers.

Table 6: Appending Different Model Architectures on Top of BERT Attention Layers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Accuracy | F1 Score | Precision | Recall | Dropout | Steps |
| **BERT Default** | **0.807** | **0.776** | 0.761 | **0.793** | 0.1 | 850 |
| BERT + Hidden Layer | 0.805 | 0.769 | **0.766** | 0.774 | 0.2 | 750 |
| BERT + Hidden Layer | 0.792 | 0.755 | 0.751 | 0.760 | 0.3 | 800 |
| BERT + 2 Hidden Layers | 0.803 | 0.768 | 0.762 | 0.776 | 0.2 | 750 |
| BERT + 2 Hidden Layers | 0.806 | 0.774 | 0.762 | 0.786 | 0.3 | 800 |
| BERT + Bi-LTSM Layer | 0.797 | 0.764 | 0.748 | 0.783 | 0.2 | 600 |
| BERT +Bi-LTSM Layer | 0.801 | 0.765 | 0.761 | 0.771 | 0.3 | 750 |
| BERT + Stacked Bi-LTSM | 0.797 | 0.764 | 0.748 | 0.781 | 0.2 | 500 |
| BERT + Stacked Bi-LTSM  **A dropout of 0.1 was not attempted on any of the new models as (Adam Thorne, 2019) demonstrated that added hidden layers with 0.1 dropout probability overfit.** | 0.80 | 0.767 | 0.753 | 0.780 | 0.3 | 700 |

In the simplest model addition of one extra hidden layer, increasing dropout probability beyond 0.2 harms performance. However, on the far more complex Bi-LTSM and + 2 hidden layer architectures, increasing dropout aids performance. It is likely that these more complicated model designs are much more prone to overfitting and so increased dropout probability is a good remedy for it.

In the end, none of the additions to the BERT fine-tuning architecture have surpassed the initial setup of a single layer for classification. Although much like the authors of [76], this paper still believes there is a logical flow to the strategy of adding a more complex architecture to interpret the BERT embeddings, especially with bi-LTSM layers which incorporate bidirectionality much like BERT does in pre-training. Perhaps more dropout is necessary in the bi-LTSM layers or further optimizations like adding an attention layer or using less hidden neurons with the bi-LTSM layers to reduce overfitting.

### 5.4.4 L2 Regularization to Normalise the Pooled [CLS] Embedding

As discussed in [Section 4.3.3](#L2reg), the biased embedding of the pooled [CLS] token can lead to degraded performance when using BERT for text classification, as BERT embeddings were pre-trained on MLM and NSP[[47]](#footnote-47), the latter task being irrelevant to text classification. The embeddings were L2 normalised on the further pre-trained model to reduce the variance in the bias distribution, but this technique was found to be in fact deleterious to model performance in terms of F1 score, although slightly. The hypothesis was that the findings in this study perhaps didn’t correlate with the findings of [79] because the further pre-training in this study was without the next sentence prediction task and so the [CLS ] embedding did not have a much of a bias towards the NSP task in the embedding distribution. Therefore, out of curiosity, L2 normalisation was executed on the model before further pre-training to see if this hypothesis was true ([Table 7](#table7)).

Table 7: The Effect of Performing L2 Normalization Before and After Further Pre-Training

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | BERT Baseline | | BERT + FPT | |
|  | Accuracy | F1 Score | Accuracy | F1 Score |
| **Not Normalised** | 0.791 | 0.752 | 0.807 | **0.776** |
| L2 Normalised | 0.787 | 0.748 | **0.808** | 0.774 |

The results in fact found that performing L2 normalisation before further pre-training was still harmful to BERT embeddings and was in fact more detrimental to performance than L2 normalisation on the embeddings after further pre-training.

# 6 Results on Test Sets

## 6.1 HatEval Dataset

In previous sections, the extensive evaluations on the training set informed the decisions to build this model. Throughout testing, the first two stages of the model (text pre-processing and further pre-training) remained consistent with these best identified practices. A single layer classification layer + focal loss was identified as the best practice in [Section 5.4](#finetuneeval). Upon initial testing however, it became apparent early that there was a significant disproportionality between the recall and precision metrics when using this method on the testing set (0.972 and 0.485 respectively). This substantial asymmetry was not forecasted in the comprehensive evaluation of the training set in previous sections, if anything the addition of focal loss when used on the HatEval dataset was forecasted to improve the precision of the classifier in comparison to the baseline cross entropy loss function. This contrast in metrics between the evaluation on the training set and the testing set suggested that there is a considerable dissimilarity between these datasets. [Section 7.1](#erroranalysis) explores this in more detail.

Improved results were then achieved on this set’s English hate speech task by using a weighted loss function to correct for a disproportionate recall in comparison to precision. More of a weighting was given to negative samples than positive, to raise the precision of the classifier at the expense of recall which had already an exorbitantly high score of 0.972 with a Focal loss implementation.

Table 8: Comparison of Different Methods in Development and Testing HatEval Set

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | | F1 Score | | Precision | | Recall | |
|  | Dev | Test | Dev | Test | Dev | Test | Dev | Test |
| Cross-Entropy Loss | 0.803 | 0.545 | 0.770 | 0.643 | 0.757 | 0.479 | **0.786** | **0.977** |
| Focal Loss | **0.806** | 0.555 | **0.773** | 0.650 | 0.764 | 0.492 | 0.784 | 0.958 |
| Weighted Loss 2:1 | 0.806 | **0.572** | 0.769 | **0.654** | **0.772** | **0.495** | 0.768 | 0.961 |
| Weighted Loss 3:1 | - | 0.560 | - | 0.652 | - | 0.485 | - | 0.975 |

**Weighted loss 2:1 was found to be the best method for correcting the imbalance between precision and recall. The ratio of 2:1 represents the weighting given to negative and positive samples respectively**

The participant systems for the HatEval competition were judged upon a macro F1 score which is different than the micro F1 score used throughout this project. Macro-averaging gives equal weight to each class, whereas micro-averaging gives equal weight to each per-document classification decision. Because the F1 measure ignores true negatives and its magnitude is mostly determined by the number of true positives, large classes dominate small classes in micro-averaging…. Micro-averaged results are therefore really a measure of effectiveness on the large classes in a test collection. To get a sense of effectiveness on small classes, you should compute macro-averaged results [85].

When a macro F1 score is calculated for the best identified system (weighted loss 2:1), the score is 0.547, which places third on the leaderboard on the test set currently on the HatEval dataset, first among the described neural network based approaches[[48]](#footnote-48) ([Table 9](#hatevealrank))

Table 9: Rankings of HatEval 2019. English Hate Speech Sub-Task

**Evaluation results as reported by official shared task spreadsheet:** [**https://docs.google.com/spreadsheets/d/1wSFKh1hvwwQIoY8\_XBVkhjxacDmwXFpkshYzLx4bw-0/edit#gid=394446410**](https://docs.google.com/spreadsheets/d/1wSFKh1hvwwQIoY8_XBVkhjxacDmwXFpkshYzLx4bw-0/edit#gid=394446410)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rank | Team | Accuracy | Macro F1 Score | System Description |
| 1 | Fermi | 0.653 | 0.651 | Universal Encoder Embeddings + SVM with RBF Kernel |
| 2 | Panaetius | 0.572 | 0.571 | No Description provided |
| **3** | **Fionn** | **0.572** | **0.547** | **BERT Embeddings + Feed-Forward Classification Layer with Weighted Loss Function** |
| 4 | YNU\_DYX | 0.560 | 0.546 | Fasttext Word Embedddings + BiGRUs + Capsule Network |
| 5 | alonzorz | 0.558 | 0.535 | Contextualized Embeddings + Multiple Choice CNN |
|  | …… |  |  |  |
| 36 | Baseline | 0.492 | 0.451 | TF-IDF embeddings + SVM |

## 6.2 AnalyticsVidhya.com Dataset

As the analyticsvidhya.com test set is completely unsupervised, one must submit predictions to the competition’s website in order to receive an evaluation. The only metric returned is an F1 Score.

Table 10: Comparison of Different Methods in Development and Testing AnalyticsVidhya.com Set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Dev | | Test | |
|  | F1 Score | Steps | F1 Score | Test |
| Cross-Entropy Loss | 0.758 | 5200 | 0.817 | 6250 |
| Focal Loss | 0.780 | 2400 | **0.827** | 2900 |
| Class-Balanced Weighted Loss | 0.785 | 2100 | 0.825 | 2500 |
| Random Oversampling | **0.793** | 2200 | 0.807 | 2650 |

As in development the training set was usually split 80:20, when testing the number of steps was increased by 20% to proportionally account for the extra data when training the model on the entire training set. This practice was likewise followed when testing using the HatEval set.

As of writing this system currently places 10th in the ongoing hackathon out of ~1000 participant systems ([Table 11](#fig20leaderboard)). Other participant approaches have not been disclosed.

Table 11: AnalyticsVidhya.com Leaderboard as of May 24th 2020

|  |  |
| --- | --- |
| Participant | F Score |
| pagadi | 0.8644 |
| robertvici | 0.8637 |
| kysmet | 0.85839 |
| becnic | 0.8575 |
| saurabh | 0.8571 |
| vasudev13 | 0.8499 |
| revathjay | 0.8466 |
| m0baxter | 0.8312 |
| vikassignh1996 | 0.8275 |
| **fionn49** | **0.8267** |

# 7 Error Analysis

An error report is only available upon the HatEval set as the Analytics.com set is unsupervised and so insights upon this dataset cannot be made.

## 7.1 Non-identically Distributed Data

When evaluating upon the training set, there was often a slight inferiority in precision when compared to recall, yet this in-depth evaluation did not forecast the extreme contrast that was found when evaluating on the test set. This was an interesting point of research so the abundance of false positive predictions of this classifier was investigated. [Table 12](#fig20) shows the top 10 most common words among false positive detections, (excluding stopwords).

Table 12: Most Common Words Among False Positive Detections

|  |  |  |
| --- | --- | --- |
| Word | Count | % of tweets containing word |
| User | 654 | 54.7 |
| Wall | 567 | 35.7 |
| Bitch | 511 | 38.2 |
| Build | 511 | 35.0 |
| Multiuser | 325 | 20.6 |
| Maga | 261 | 20.3 |
| Daca | 222 | 15.5 |
| Trump | 212 | 16.2 |
| Illegal | 192 | 12.4 |
| America | 190 | 15.4 |
| Hoe | 142 | 21.6 |

Upon further inspection of the test set, it became apparent that there was a significant disparity between the training set and the test set for the words “bitch”, “maga” and the phrase “build the wall”. The contrast between the training set and the test set can be observed by comparing the ratios of the distribution of these terms in hate speech and non-hate speech subsets ([Table 13](#table9)).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Train | | | |  | Test | | | |
|  | HS | ¬HS | Ratio | **Total** |  | HS | ¬HS | Ratio | **Total** |
| “bitch” | 22.6% | 4.7% | 4.8:1 | **12.2%** |  | 50.2% | 47.7% | 1..1:1 | **48.7%** |
| “build the wall” | 5.9% | 0.3% | 19.7:1 | **2.7%** |  | 11.4% | 6.6% | 1.7:1 | **8.6%** |
| “maga” | 4.8% | 0.7% | 6.9:1 | **2.6%** |  | 8.5% | 15.5% | 0.55:1 | **12.5%** |

Table 13: Percentage of Tweets Containing Each Term

One of the most common assumptions in many machine learning and data analysis tasks is that the given data points are realizations of independent and identically distributed (IID) random variables [85]. This assumption is rarely satisfied in practice, however great care should be taken to realise identically distributed data when training a classifier that automatically detects hate speech. As discussed in [Section 2.1](#LimitationsHSdata), many hate speech detectors are prone to false positives, precisely because the data that they are trained upon is oversensitive to the occurrence of these terms in text, whether they are terms that are commonly used to disparage (“bitch) or they are politically contentious (“build the wall”, “maga”).

It is critical that a more equal distribution of non-hate speech and hate speech tweets containing these controversial terms exist in the training data, as an automatic classifier’s greatest difficulty when classifying hate speech is ascertaining the contextual relationship between these controversial terms and the text. In real-world application, these slurs and politically controversial utterances are in abundance online, often in a non-hateful context and so automatic classifiers vitally must learn this distinction in training.

Furthermore, the disparity between the HatEval training set and the testing set makes it difficult to infer a reliable measure of the model’s capability of generalising well. The model predicts the rallying cries of Trump acolytes (“build the wall”, “maga”) as hate speech every time it is in a tweet ([Table 14](#fig10)) because it is very rarely exposed to instances of the contrary in training[[49]](#footnote-49). The false positive misclassifications with these terms may not reflect the classifier’s poor learning ability, but rather it reflects the distribution of tweets containing these terms in the data.

Table 14: Model Predictions on Test Set When a Term is Present in the Tweet

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Hate Speech Tweets | | |  | Non-Hate Speech Tweets | | |
|  |  | | |  |  | | |
|  | Predictions | Ground Truth | **% Misclassified** |  | Predictions | Ground Truth | **% Misclassified** |
| “bitch” | 607 | 632 | **3.96%** |  | 353 | 830 | **57.5%** |
| “build the wall” | 144 | 144 | **0%** |  | 0 | 115 | **100%** |
| “maga” | 107 | 107 | **0%** |  | 0 | 267 | **100%** |

Despite quite advanced machine learning architectures attempting to tackle this dataset, relatively poor scores for a binary classification task have been achieved so far, with the best score being achieved by an SVM classifier. The authors of the shared task based upon this dataset [34] have highlighted that the three best performing systems are highly skewed towards false positive detections and predict the same false positive labels 507 times out of 2,971 total instances (17.1% of the time). 89.1% of the 569 in-common errors that each of the three best performing systems made were false positives. The contrasting distribution of these terms in the training set and the test set may be the cause of this.

## 7.2 False Positive Annotations

Alongside the reasons for an abundance in false positive annotation discussed in [Section 7.1](#nonidenticaldistributeddata), the classifier struggled with offensive language being used in a sarcastic or humorous manner, a common problem among many NLP applications [86]. This, coupled with tweets including feminine-based slurs, especially when these slurs appeared in the text multiple times, a false positive classification for hate speech often occurred ([Figure 20](#fig22)).

Figure 20: False Positive Tweets with Feminine Based Slurs

|  |  |
| --- | --- |
| ID | Tweet |
| 31881 | i did not come to play with you hoes I came to slay bitch |
| 33022 | i dont love you bitch you a hoe I never will cant tell you my feelings because pimpin dont feel user |
| 31048 | happy birthday bitch you not a teenager no more hoe its about damn time fuck it up like you been fucking it up |

Likewise, when there was a high content of trump-related content this seemed to often trigger a false positive detection. Whilst one could argue that some of the following tweets have unscrupulous content, they fall short of the definition of hate speech that guides this project and they fail to fit the definition of hate speech according to a majority opinion of annotators ([Figure 21](#fig23))

Figure 21: False Positive Tweets with Immigrants as the Subject

|  |  |
| --- | --- |
| ID | Tweet |
| 34032 | user just received my copy of mad politics looking forward to a mad politics weekend your great perspective dr gina maga kag potus trump news vote red no daca walk away america first usa back the blue all lives matter drain the swamp liberalism is a mental disorder |
| 33130 | trump says he could use military to build his wall if congress wont fund it though homeland securitys budget and he wont rule out another government shutdown to get his way whatever it takes user build that wall |
| 3240 | Imultuser mad max is whined up doll the same saying is repeated does she even care that the american people are striving with our booming economy maga drain the swamp build that wall |

On a side-note however, many tweets were observed in the false positive set that would fulfil the definition of hate speech guiding this project ([Figure 22](#fig24)). The tweet containing the phrase “anchor babies” may have been mislabelled by annotators unaware of the pejorative sentiment of the phrase. Despite president Trump using the phrase to describe a political rival [87], the phrase should not be accepted in the public discourse and flagged as hate speech if used in a pejorative context [88].

Figure 22: Possible HS Tweets That Were Labelled as ~HS

|  |  |
| --- | --- |
| ID | Tweet |
| 33255 | multiuser this refers 2 a favourable political effort by google 2 dem partys pres candidate they were surprised that sure fire voters they gave rides 2 didn’t all vote 4 their woman they expected it like the blacks used 2 do but begin to wake up next up muslims stop the invasion |
| 30674 | theyre called anchor babies than you user build the wall send them back |
| 34393 | multiuser thanks for allowing violent foreign men to walk among us threaten us because of user law court system favours them over decent law abiding people civil war is looming in their country as you do nothing to help or protect the people send them back |
| 31048 | user of Europe that’s not a humanitarian ship that’s oros they must go back stop the invasion |

The phrase “send them back” also has racist/xenophobic connotations, especially in the tweets above. It is an “othering” phrase which is an established construct in the rhetorical narrative surrounding hate speech [89]. The “we-they” or “us-them” dichotomy has long been identified as a characteristic in racist discourse [90]. The phrase is so unconscionable that even Mr. Trump distanced himself from it when a rally of his supporters chanted a similar idiom of ‘send her back’ in response to his scathing criticism of a Muslim congresswoman [91]. Unless this phrase is used in a fashion where the subject is critical of the phrase, it should be regarded as hate speech.

## 7.3 False Negative Annotations

False negative errors were quite rare in evaluation upon the test set[[50]](#footnote-50), only 47 tweets out of a possible 1260 hate speech tweets were missed by the classifier, a false negative rate of 3.73%.

The false negative tweets below are hateful tweets directed at migrants but phrased in a relatively non-aggressive manner which is possibly why they were missed by the classifier. They are devoid of the traditional slurs normally associated with hate speech, which may make identifying hateful content easier for classifiers. ([Figure 23](#fig25))

Figure 23: False Negative Misclassifications

|  |  |
| --- | --- |
| ID | Tweet |
| 31978 | there will be Cronulla roits 2 in Melbourne we are sick of it send them home |
| 30950 | user sea tow the boat back to the safe ports in africa send them back its not late save Europe not wanted not needed |
| 31373 | animals pure and simple they are not to blame for where they were born or how they were raised but as adult men who have escaped their homeland they alone are responsible for how they act during each every moment after crossing the border send them back |

Also, semantically positive words like “safe”, “save” and “responsible” may have thrown off the classifier.

# 8 Discussion

In this project, a convincing argument is made for careful pre-processing of tweet data. One should not dispose of emojis or leave hashtags to be processed by the embedding space untouched. Useful information is contained within these features of tweet text and if processed properly they can advise a classifier on the sentiment of a tweet, whether it is malintent, sarcasm or humour. Adding tokens to represent emojis in the BERT vocab file proved to be less beneficial to converting them to words. Even with the addition of further pre-training which allowed these tokens to gain an improved weighting in the embedding space it still fell short of translating the emojis to words.

Tweet datasets are often sourced using an in-common term as mentioned in [Section 4.2](#FurtherPretrainstage). Further pre-training with these corpora elicits concerns for the learning of the MLM task in further pre-training, as it can cause the training process to over-predict these terms. Removing these terms for the data used in further pre-training however, results in a loss of semantic meaning for the tweets and is proven to be much more deleterious to natural language understanding as it decisively performs worse than not removing the terms. This gives insight into how robust the MLM task is to terms that appear frequently and how vital semantic meaning is to the pre-training of BERT. Further insight into the optimal amount of steps and the best initial learning rate were made, however not enough to reach a consensus, additional steps for further pre-training could be experimented with in future studies, however the performance gain this extra effort and experimentation extract seems slight and there is no decisive confirmation that these parameters are generally applicable, they may be task specific.

Much like the authors of [76], a successful method of appending a more complex neural architecture to BERT was not forthcoming. With more research, skill and with better tuning of parameters however, this ambition could be realised. The addition of different loss functions proved successful, there are significant performance benefits in using these functions rather than the default categorical cross entropy loss function that the authors of BERT originally implemented. This approach should generally work for most applications that require a cost-sensitive approach, whether it is the prioritisation of retaining positive identifications for a class or if it is managing an imbalanced classification problem.

The high interest of the open source community in hate speech detection is evident by the large number of participants and enthusiasm for shared tasks and competitions covering this subject. Several papers have been written and many have deliberated the best approach for designing a system that accurately detects hate speech. Unfortunately, due to the lack of reliably annotated hate speech data online, the relative shortage of such data and an absence of a consistent interpretation of what constitutes hate speech between these corpora - it is presently not possible for the open source community to develop a hate speech classifier that is qualified for commercial use. The classifier developed here and indeed any classifier that is trained upon available data would not be fit for real world purpose. A perfect classifier is not expected, however too often does this classifier label a tweet as hate speech due to an insensitivity to AAE dialect (resulting in racial bias) or due to the appearance of a term associated with Mr. Trump[[51]](#footnote-51). This is caused more by the flaws in the training and testing data itself rather than by the ability of the model to generalise well. In a perfect world, these companies would release the vast amount of hate speech data they have surely collected and allow the research community to collaborate on a solution to the scourge of online hate speech. For now however, the online community have presently no choice but to rely upon the social media giants of Facebook, YouTube and Google to police online discourse with their respective proprietary systems.

9 References

|  |  |
| --- | --- |
| [1] | https://gitlab2.eeecs.qub.ac.uk/csc3002\_fionn/csc3002\_detecting\_hate\_speech. |
| [2] | J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” 2019. |
| [3] | Economist, “Why free speech, hate speech and radicalisation are hard to define,” *Economist,* 1 Febuary 2019. |
| [4] | D. Alba, “Defining 'Hate Speech' Online Is an Imperfect Art,” *Wired,* 22 9 2017. |
| [5] | K. Rawlinson, “Finsbury Park-accused trawled far-right groups online, court told,” *The Guardian,* 23 January 2018. |
| [6] | F. Mao, “Christchurch shooting: Australia's moment of hate speech reckoning,” *BBC Newa,* 20 March 2019. |
| [7] | M. Al-Youssef, “Government Seeks to Eliminate Internet Anonymity – With Severe Penalties,” *Der Standard,* 18 April 2019. |
| [8] | J. C. York, “Facebook's 'real names' policy is legal, but it's also problematic for free speech,” *The Guardian,* 29 September 2014. |
| [9] | J. W. Howard, “Free Speech and Hate Speech.,” *Annual Review of Political Science,* pp. 93-109, 2019. |
| [10] | U. S. Court, “Brandenburg v. Ohio,” 1969. |
| [11] | A. Oboler, “ Online Hate Prevention Institute,” 2014. |
| [12] | D. W. Sue, C. M. Capodilupo, G. C. Torino, J. M. Bucceri, A. Holder, K. L. Nadal and M. Esquilin, “Racial microaggressions in everyday life: implications for clinical practice,” *American Psychologist,* pp. 271-286, 2007. |
| [13] | U. f. I. Action, “STEP IN! BE ACTIVE AGAINST RACIST PROPAGANDA AND HATE SPEECH ONLINE,” in *UNITED for Intercultural Action*, Strasbourg, 2012. |
| [14] | J. T. Nockleby, “Hate Speech,” *Encyclopedia of the American Constitution (2nd ed),* pp. 1277-1279, 2000. |
| [15] | E. S. Olivas, J. D. M. Guerrero, M. M. Sober, J. R. M. Benedito and A. J. S. Lopez, Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques, IGI Publishing, 2009, 2009. |
| [16] | A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. P. , “Attention Is All You Need,” 2017. |
| [17] | M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, “Deep contextualized word representations,” NAACL, 2018. |
| [18] | A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, “Improving language understanding with unsupervised learning,” OpenAI, 2018. |
| [19] | Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba and S. Fidler, “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” in *IEEE international conference on computer vision*, 2015. |
| [20] | Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach.,” 2019. |
| [21] | Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov and Q. V. Le, “XLNet: Generalized Autoregressive Pretraining for Language Understanding,” 2019. |
| [22] | E. Greevy and A. F. Smeaton, “Classifying racist texts using a support vector machine,” in *27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004. |
| [23] | W. Warner and J. Hirschberg, “Detecting hate speech on the world wide wb,” in *Second Workshop on Language in Social*, 2012. |
| [24] | C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad and Y. Chang, “Abusive Language Detection in Online User Content,” in *25th International Conference on World Wide Web*, 2016. |
| [25] | P. Badjatiya, S. Gupta, M. Gupta and V. Varma, “Deep learning for hate speech detection in tweets,” in *26th International Conference on World Wide Web Companion*, 2017. |
| [26] | Z. Zhang, D. Robinson and a. J. Tepper, “Detecting hate speech on twitter using a convolution-gru based deep neural network,” in *European Semantic Web Conference*, 2018. |
| [27] | K. Cho, B. v. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, “Learning phrase representations using rnn encoder–decoder for statistical machine translation,” in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. |
| [28] | Z. Waseem, W. Chung, H. Kyong, D. Hovy and J. Tetrault, in *First Workshop on Abusive Language Onlin*. |
| [29] | T. Davidson, D. Warmsley, M. Macy and I. Weber, “Automated Hate Speech Detection and the Problem of Offensive Language,” in *Eleventh International AAAI onference on Web and Social Media*, Montreal, 2017. |
| [30] | A.-M. Founta, D. Constantinos, C. Despoina, I. Leontiadis, J. Blackburn, G. Stringhini and A. V. Michael, “Large scale crowdsourcing and characterization of twitter abusive behavior,” in *AAAI*, 2018. |
| [31] | F. Poletto, M. Stranisci, M. Sanguinetti, V. Patti and C. Bosco, “Hate Speech Annotation: Analysis of an Italian Twitter Corpus.,” in *In Proceedings of the Fourth Italian Conference on Computational Linguistics*, 2017. |
| [32] | E. Fersini, P. Rosso and M. Anzovino, “Overview of the Task on Automatic Misogyny Identification at IberEval 2018,” in *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages*, 2018. |
| [33] | M. Wiegand, M. Siegel and J. Ruppenhofer, “Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language,” in *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing*, 2018. |
| [34] | V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. Rangel, P. Rosso and M. Sanguinetti, “Semeval2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter,” in *13th International Workshop on Semantic Evaluation (SemEval-2019)*, 2019. |
| [35] | V. Indurthi, B. Syed, M. Shrivastava, N. Chakravartula, M. Gupta and V. Varma, “Fermi at SemEval-2019 Task 5: Using Sentence Embeddings to identify Hate Speech against Immigrants and Women on Twitter,” in *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*, 2019. |
| [36] | D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope and R. Kurzweil, “Universal Sentence Encoder,” 2018. |
| [37] | A. Joulin, E. Grave, P. Bojanowski and T. Mikolov, “Bag of tricks for efficient text classification,” in *15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2*, 2017. |
| [38] | S. Sabour, N. Frosst and a. G. E. Hinton, “Dynamic routing between capsules,” in *Advances in neural information processing systems*, 2017, p. 3856–3866.. |
| [39] | M. Zampieri, S. Nakov, P. Rosenthal, S. Farra, N. Kumar and R. Malmasi, “Predicting the Type and Target of Offensive Posts in Social Media,” in *NAACL*, 2019. |
| [40] | Z. Waseem and D. Hovy, “Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter,” 2016. |
| [41] | T. Davidson, D. Bhattacharya and I. Weber, “Racial Bias in Hate Speech and Abusive Language Detection Datasets,” 2019. |
| [42] | J. Rahman, “The N word: Its history and use in the African American Community,” 2012. |
| [43] | I. Kwok and Y. Wang, “Locate the Hate: Detecting Tweets against Blacks,” in *In Association for the Advancement of Artificial Intelligence*, 2013. |
| [44] | P. Burnap and M. L. Williams, “Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making,” *Policy & Internet,* pp. 223-242, 2015. |
| [45] | L. Dixon, J. Li, J. Sorensen, N. Thain and L. Vasserman, “Measuring and Mitigating Unintended Bias in Text Classification,” in *AAAI/ACM Conference on AI, ethics and society*, New Orleans, 2018. |
| [46] | M. Sap, D. Card, S. Gabriel, Y. Choi and N. A. Smith, “The Risk of Racial Bias in Hate Speech Detection,” 2019. |
| [47] | T. Lapin, “Facebook flagged Declaration of Independence as hate speech,” *New York Post,* 2018. |
| [48] | Z. Leonard, I Want a President, https://www.serpentinegalleries.org/files/downloads/i\_want\_a\_president\_original\_0.pdf, 1992. |
| [49] | S. Wolfson, “Facebook labels declaration of independence as 'hate speech',” *The Guardian,* 5 July 2018. |
| [50] | J. H. Park, J. Shin and P. Fung, “Reducing Gender Bias in Abusive Language Detection,” in *Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018. |
| [51] | J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So and J. Kang, “BioBERT: a pre-trained biomedical language representation model for biomedical text mining,” in *Bioinformatics, Volume 36, Issue 4,*, 2020, p. 1234–1240. |
| [52] | E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann and M. B. A. McDermott, “Publicly Available Clinical BERT Embeddings,” 2019. |
| [53] | C. Sun and X. Qiu, “How to Fine-Tune BERT for Text Classification?,” Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China, 2020. |
| [54] | A. Blasdel, “How the resurgence of white supremacy in the US sparked a war over free speech,” *Guardian,* 31 May 2018. |
| [55] | J. Berger, “Trump Is the Glue That Binds the Far Right,” *The Atlantic,* 29 October 2019. |
| [56] | E. Summers, “UniteTheRight Tweet IDs,” https://archive.org/details/unitetheright-ids, 2017. |
| [57] | M. Keneally, “What to know about the violent Charlottesville protests and anniversary rallies,” *abcNEWS,* pp. https://abcnews.go.com/US/happen-charlottesville-protest-anniversary-weekend/story?id=57107500, 8 August 2018. |
| [58] | B. Jules, “#Charlottesville,” UC Riverside, https://doi.org/10.6086/D10T0B, 2017. |
| [59] | B. Jules, “#BLMKidnapping,” https://doi.org/10.6086/D1CC7X, 2017. |
| [60] | H. Yan and C. Zdanowicz, “Police: No link between Chicago torture video and Black Lives Matter,” *CNN,* 5 January 2017. |
| [61] | N. Doiron, “Replies to Ocasio-Cortez Tweets,” https://github.com/mapmeld/aoc\_reply\_dataset, 2019. |
| [62] | N. Ruest, “#thechalkening tweet dataset,” https://dataverse.scholarsportal.info/dataset.xhtml?persistentId=hdl:10864/11591, 2016. |
| [63] | S. Tesfaye, “"The Chalkening" has gone too far: Trump's call for college supporters to arm themselves with chalk leaves campuses littered with racist messages,” *Salon,* 11 April 2016. |
| [64] | P. Binkley, “Alberta Bill 10 twitter IDS,” https://doi.org/10.7939/DVN/10971, 2017. |
| [65] | D. Kinney, “Alberta Should Be Proud Of Gay-Straight Alliances In Its Schools,” *Huffington Post Cnada,* 1 June 2017. |
| [66] | M. E. Phillips, “"Yes All Women" Twitter Dataset,” University of North Texas Libraries, UNT Digital Library, https://digital.library.unt.edu/ark:/67531/metadc304853/, 2014. |
| [67] | J. Zimmerman, “Not All Men: A Brief History of Every Dude's Favorite Argument,” *Time,* 28 April 2014. |
| [68] | M. Martin and A. Chu, “Jeopardy Champ Arthur Chu On Nerds, Entitlement And Elliot Rodger,” *NPR,* 30 May 2014. |
| [69] | S. Weiss, “The Power of #YesAllWomen,” *The New Yorker,* 2016 May 2014. |
| [70] | J. Littman, “Immigration and Travel Ban Tweet Ids,” https://doi.org/10.7910/DVN/5CFLLJ, Harvard Dataverse, V1, 2018. |
| [71] | M. D. Shear and H. Cooper, “Trump Bars Refugees and Citizens of 7 Muslim Countries,” *The New York Times,* 27 January 2017. |
| [72] | D. Lee, “Twitter's hate speech rules are expanded,” *BBC News,* 18 December 2017. |
| [73] | P. Norvig, “Natural Language Corpus Data,” in *Beautiful Data*, Segaran and Hammerbacher,, 2009. |
| [74] | Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao and K. Macherey, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” arXiv preprint arXiv:1609.08144., 2016. |
| [75] | M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in *Psychology of learning and motivation, volume 24*, 1989, pp. 109-165. |
| [76] | A. Thorne, Z. Farnsworth and O. Matus, “Layers, Research of LSTM Additions on Top of SQuAD BERT Hidden Transform,” Stanford, 2019. |
| [77] | T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollar, “Focal Loss for Dense Object Detection,” Facebook AI Research (FAIR), 2018. |
| [78] | Y. Shi, J. Meng and J. Wang, “A Normalized Encoder-Decoder Model for Abstractive Summarization Using Focal Loss,” in *Natural Language Processing and Chinese Computing*, 2018. |
| [79] | X. Sun, K. Dong, L. Ma, R. S. F. He, S. Chen and J. Feng, “Drug-Drug Interaction Extraction via Recurrent Hybrid Convolutional Neural Networks with an Improved Focal Loss,” *entropy,* 2019. |
| [80] | W. Zhou, J. Du and X. Ren, “Improving BERT Fine-tuning with Embedding Normalisation,” University of Southern California, 2019. |
| [81] | Y. Zhang, Y. Che, S.-C. Cheung, Y. Xiong and L. Zhang, “An Empirical Study on TensorFlow Program Bugs,” in *27th ACM SIGSOFT International Symposium on Software Testing*, 2018. |
| [82] | H. B. Moss, D. S. Leslie and P. Rayson, “Using J-K-fold Cross Validation to Reduce Variance When Tuning NLP Models,” 2018. |
| [83] | A. Saif, H. Hassan, Miriam, He and Y. Fernández, “On stopwords, filtering and data sparsity for sentiment analysis on twitter,” in *LREC 2014, Ninth International Conference on Language Resources and Evaluation*, 2014. |
| [84] | Tensorflow, “tf.nn.weighted\_cross\_entropy\_with\_logits,” p. https://www.tensorflow.org/api\_docs/python/tf/nn/weighted\_cross\_entropy\_with\_logits, 2020. |
| [85] | C. D. Manning, P. Raghavan and H. Schutze, Introduction to Information Retrieval, Cambridge, UK: Cambridge University Press, 2008. |
| [86] | T. Darrell, M. Kloft, M. Pontil, G. Rätsch and E. Rodner, “Machine Learning with Interdependent and Non-identically Distributed Data (Dagstuhl Seminar 15152),” *Dagstuhl Reports, Volume 5,* pp. 18-55, 2015. |
| [87] | H. Heidenreich, “NLP and Sarcasm: What's the Deal?,” *http://hunterheidenreich.com/blog/nlp-and-sarcasm/,* 13 Febuary 2019. |
| [88] | J. Diamond, “Donald Trump: Ted Cruz is an 'anchor baby',” *CNN,* 30 January 2016. |
| [89] | R. L. Contreras, “‘Anchor babies’ is hate speech,” *San Diego Union Tribune Logo,* 24 August 2007. |
| [90] | K. J. Meddaugh PM, “Hate speech or ‘reasonable racism?’ The other in stormfront. J Mass Media Ethics,” 2009. |
| [91] | W. R and R. M, in *Discourse and racism: European perspectives*, 1999, pp. 175-199. |
| [92] | A. Zurcher, “Trump - I disagree with Omar 'send her back' chants,” *BBC,* 18 July 2019. |
| [93] | Z. Waseem, “Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter,” 2016. |
| [94] | J. Jolly, “Ferguson riots: Ruling sparks night of violence,” *BBC News,* 25 November 2014. |
| [95] | A. Villarreal, “Anchor babies': the 'ludicrous' immigration myth that treats people as pawns,” *The Guardian,* 16 March 2020. |
| [96] | W. P. Burnap, “Cyber hate speech on Twitter: an application of machine classification and statistical modeling for policy and decision making.,” in *Policy & Internet*, 2015, pp. 223-242. |
| [97] | W. Alorainy, P. Burnap, H. Liu, Amir Javed and M. L. Williams, “Suspended Accounts: A Source of Tweets with Disgust and Anger Emotions for Augmenting Hate Speech Data Sample,” in *International Conference on Machine Learning and Cybernetics (ICMLC)*, Chengdu, 2018. |

1. The version of BERT used throughout this study was BERT-Large, Uncased (Whole Word Masking) which is a 24-layer, 1024-hidden, 16-heads, 340M parameters pre-trained model [↑](#footnote-ref-1)
2. https://datahack.analyticsvidhya.com/contest/practice-problem-twitter-sentiment-analysis/#About [↑](#footnote-ref-2)
3. This is otherwise known as the Brandenburg test [↑](#footnote-ref-3)
4. Estimates are around 500 million tweets a day are posted [↑](#footnote-ref-4)
5. Which stands for Bidirectional Encoder Representations fromTransformers [↑](#footnote-ref-5)
6. 1M steps in pre-training on an initial learning rate of 5e-5 [↑](#footnote-ref-6)
7. The [CLS] token is the last token in a tweet/sentence. [↑](#footnote-ref-7)
8. Emojis are pre-processed appropriately before being used to further pre-train embeddings [↑](#footnote-ref-8)
9. It is hypothesised that translating to words before pre-training is advantageous because words have already pre-existing weights in the embedding space as they have been seen already in the extensive pre-training that BERT has underwent. The emoji tokens which are added to the vocab however have not had this luxury. Their weights are effectively randomly initialized and only build upon in a relatively transient fine-tuning stage. [↑](#footnote-ref-9)
10. http://evalita.org [↑](#footnote-ref-10)
11. http://sites.google.com/view/ibereval-2018 [↑](#footnote-ref-11)
12. The second-best performing team did not provide a description of their system [↑](#footnote-ref-12)
13. Minimum 3 annotators per tweet, so a 2-person majority at minimum was needed [↑](#footnote-ref-13)
14. https://hatebase.org/ [↑](#footnote-ref-14)
15. The authors of this dataset clearly state that the entry `hateful` is meant to represent hate speech and they have instructed their annotators to label these tweets on that basis. [↑](#footnote-ref-15)
16. Tweets extracted from twitter using Tweepy API. [↑](#footnote-ref-16)
17. The expert annotators for much of the dataset were “a 25-year-old woman studying gender studies and a nonactivist feminist”, based upon the assumption that they are domain-experts. Their annotation was further reviewed for around 3K out of the total 17K tweets by “feminist and anti-racism activists” [↑](#footnote-ref-17)
18. https://datahack.analyticsvidhya.com/contest/practice-problem-twitter-sentiment-analysis/#ProblemStatement [↑](#footnote-ref-18)
19. https://www.perspectiveapi.com [↑](#footnote-ref-19)
20. The filter was triggered by the declaration using the slur “Indian savages” which would be hate speech in most contexts. [↑](#footnote-ref-20)
21. The hypothesis for the OffensEval 2019 dataset, whereby if one combined columns that were labelled as offensive, a targeted insult and directed at a group/individual it could amount to hate speech, was unfortunately not accurate. The tweets that satisfied the conditions of in this dataset were often the result of common political quarrelling, that is unfortunately rampant in the USA right now, rather than hate speech. [↑](#footnote-ref-21)
22. Perhaps these tweets could have qualified as hate speech based upon who they were targeting. It is difficult to ascertain this from a linguistic standpoint due to the decontextualized nature of these tweets. [↑](#footnote-ref-22)
23. The ICVSM 2017 dataset was also mostly reliable (92% intercoder score) and would have been a good candidate for analysis - despite some minor flaws. [↑](#footnote-ref-23)
24. The classification of hate speech on twitter is inherently an imbalanced classification problem. [30] [↑](#footnote-ref-24)
25. The datasets are specifically targeted to be within-domain as this type of data is superior for further pre-training in comparison to cross-domain corpora and within task corpora [53] [↑](#footnote-ref-25)
26. Care was taken to avoid using test sets in further pre-training [↑](#footnote-ref-26)
27. Twitter’s terms of service do not allow tweet datasets to be published on the web, but they do allow tweet identifier datasets to be shared [↑](#footnote-ref-27)
28. https://github.com/tweepy/tweepy [↑](#footnote-ref-28)
29. Text pre-processing removed the retweet (RT) handle and made mentions a common tag so that all retweets after pre-processing were identical in textual form. This, along with tweet datasets discussing the same issue at the same time (e.g. #unitetheright and #charlottesville) meant that approximately 636,224 duplicate retweets were removed before further pre-training. [↑](#footnote-ref-29)
30. Usually via twitter API streaming using keywords or common hashtags [↑](#footnote-ref-30)
31. The “combined” set is comprised of the AOC replies, #unitetheright, #bill10, #BLMkidnapping and #charlottesville datasets [↑](#footnote-ref-31)
32. Every text pre-processing technique demonstrated in this section can be observed in practice in the Text\_Preprocessing.ipynb notebook in the Text\_Preprocessing directory of the Gitlab repo. Choices as to what pre-processing pipeline the user might elect to use can be customised in the further pre-training notebook or the fine-tuning notebook, as detailed in the README file in each respective directory [↑](#footnote-ref-32)
33. A “user” tag was used for a mention and for consecutive mentions the common tag “multiuser” was used. These were added to the BERT vocab file so they could gain slightly better word embeddings than the wordpiece tokenization would have given them [↑](#footnote-ref-33)
34. Without hashtag segmentation, #sendthemback is split into wordpieces by BERT’s inbuilt tokenizer as [‘#’, ‘send’ , ‘###the’, ‘##mba’, ‘##ck’] rather than [‘send’, ‘them’, ‘back’] [↑](#footnote-ref-34)
35. https://pypi.org/project/wordsegmentation/ [↑](#footnote-ref-35)
36. https://pypi.org/project/emoji/ [↑](#footnote-ref-36)
37. The vocab.txt file mapped a wordpiece to a word id which has an associated weighting [↑](#footnote-ref-37)
38. The current most popular emojis in real-time were extracted by parsing the HTML from the website http://www.emojistats.org/ [↑](#footnote-ref-38)
39. This was achieved by forking the official BERT repo and editing the run\_pretraining.py file to not include the loss from the NSP task and changing all segment embeddings to the same ID [↑](#footnote-ref-39)
40. Catastrophic forgetting is a common problem in transfer learning where pre-trained knowledge is erased during the learning of new knowledge [↑](#footnote-ref-40)
41. A bi-directional implementation of an LTSM [↑](#footnote-ref-41)
42. In both bi-LTSM and multi-layer perceptron implementations [↑](#footnote-ref-42)
43. Weights being based upon the ratio of positive to negative samples in training [↑](#footnote-ref-43)
44. The original implementation returned only accuracy and loss metrics [↑](#footnote-ref-44)
45. Each experiment had to be executed upon TPUs in a distributed strategy because of the sheer size of BERT-large. [↑](#footnote-ref-45)
46. However, the authors did not explicitly specify how much data further pre-training was executed upon, so it was hard to ascertain if this was at all definitive guidance. BERT’s official repo vaguely suggests further pre-training for more than 10,000 steps. [↑](#footnote-ref-46)
47. MLM = Masked Language Modelling and NSP = Next Sentence Prediction [↑](#footnote-ref-47)
48. The 2nd best performing system on the hate speech sub-task of the shared task did not provide a description of their system. [↑](#footnote-ref-48)
49. Relative to how many tweets with these terms are annotated as hate speech [↑](#footnote-ref-49)
50. Overall, false negatives made up 3.6% of errors [↑](#footnote-ref-50)
51. Even though silencing Trump devotees seems like a pleasant concept, it would not be appropriate. [↑](#footnote-ref-51)