Using Transfer-based Language Models to Detect Hateful and Offensive Language Online

Vebjørn Isaksen* and **Björn Gambäck**
Department of Computer Science
Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
vebjorni@me.com, gamback@ntnu.no

Abstract

Distinguishing hate speech from non-hate offensive language is challenging, as hate speech not always includes offensive slurs and offensive language not always express hate. Here, four deep learners based on the Bidirectional Encoder Representations from Transformers (BERT), with either general or domain-specific language models, were tested against two datasets containing tweets labelled as either 'Hateful', 'Normal' or 'Offensive'. The results indicate that the attention-based models profoundly confuse hate speech with offensive and normal language. However, the pre-trained models outperform state-of-the-art results in terms of accurately predicting the hateful instances.

1 Introduction

The majority of the tweets on Twitter or posts on Facebook are harmless and often posted purposefully, but some express hatred towards a targeted individual or minority group and members. These posts are intended to be derogatory, humiliating or insulting and are defined as hate speech by Davidson et al. (2017). Different from offensive language, hate speech is usually expressed towards group attributes such as religion, ethnic origin, sexual orientation, disability or gender (Founta et al., 2018b). Some of the biggest firms invest heavily in tracking abusive language, e.g., automatic detection of offensive language in comments (Systrom, 2017, 2018) or giving a percentage of how likely a text is to be perceived as toxic.[1] However, these and other existing tools share a common flaw of not distinguishing between offensive and hateful language. One important reason to keep these two separate is that hate speech is considered a felony in many countries. The task of separating offensive and hateful language has shown to be demanding; however, with the recent scientific breakthroughs and the concept of transfer learning, we can take huge steps in the right direction.

The paper investigates the effects of transferring knowledge from the Bidirectional Encoder Representations from Transformers (BERT; Devlin et al., 2019) language model on distinguishing hateful, offensive and normal language, by fine-tuning the pre-trained BERT language model with data containing hateful and offensive language, and comparing its performance to the state-of-the-art on two widely used hate speech detection datasets. Those datasets are presented Section 2. Section 3 then gives an overview of related work in the field of hate speech detection. Section 4 describes the implemented system architecture. Section 5 presents the experiments, including setup and results, while Section 6 evaluates and discusses those results. Section 7 concludes and suggests future work.

2 Data

Many existing datasets containing hate speech are publicly available for use and consist of data from several sources online, mainly Twitter (Waseem and Hovy, 2016; Waseem, 2016; Chatzakou et al., 2017; Golbeck et al., 2017; Davidson et al., 2017; Ross et al., 2016; ElSherief et al., 2018; Founta et al., 2018b), while some cover other sources such as Fox News comments (Gao and Huang, 2017) and sentences from posts on the white supremacist online forum Stormfront (de Gibert et al., 2018). Almost all available datasets are labelled by humans,[2] which results in different approaches taken when creating and annotating the datasets. Some researchers use expert annotators (Waseem and Hovy, 2016), others use majority voting among several

*Currently at Bekk Consulting AS

[1] https://www.perspectiveapi.com

[2] Except for the 12M tweet SOLID dataset (Rosenthal et al., 2020). It is, however, distance-learned based on the manually annotated 14k OLID tweet set (Zampieri et al., 2019).

Proceedings of the Fourth Workshop on Online Abuse and Harms, pages 16–27
Online, November 20, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

Class	Normal	Offensive	Hateful
Tweets	4,163	19,190	1,430

Table 1: The Davidson et al. (2017) dataset, **D**

Dataset	Normal	Offensive	Hateful	Spam
Original	53,790	27,037	4,948	14,024
Available	41,784	14,202	2,941	9,372

Table 2: The Founta et al. (2018b) dataset, **F**

amateur annotators on platforms such as CrowdFlower (Davidson et al., 2017). However, the task of hate speech detection lacks a shared benchmark dataset (Schmidt and Wiegand, 2017) that can be used to measure the performance of different machine learning models. Further, most annotation schemata follow Waseem and Hovy (2016) by splitting the data into only two basic classes, either hate and none hate or offensive and non-offensive (classes that then also often are split, e.g., labelling hateful tweets as either sexist or racist). However, it is debatable whether those labels are sufficient to represent hateful and abusive language. In contrast, a few datasets make the distinction between hateful and offensive language, e.g., Davidson et al. (2017) and Founta et al. (2018b), which will be used here and abbreviated **D** and **F**, respectively.

The dataset by Davidson et al. (2017) consists of 24,783 English tweets and their labels along with some information including the number of annotators. The number of CrowdFlower annotators range from three to nine, and majority voting was used when deciding the final class for a tweet: "Hate Speech", "Offensive Language" or "Neither". The label distribution can be seen in Table 1.

The dataset created by Founta et al. (2018b) contains almost 100k annotated tweets with four labels, "Normal", "Spam", "Hateful" and "Abusive". As the authors only provide tweet IDs for researches to retrieve tweets through the Twitter Application Programming Interface (API), some tweets may for several reasons not be retrievable, e.g., a tweet or the user account behind a tweet may have been deleted; thus, of the 99,799 provided tweet IDs, only 68,299 tweets were retrieved. The label distribution of those compared to the original label distribution for dataset **F** is shown in Table 2.

3 Related Work

Nobata et al. (2016) mention some challenges within hate speech, e.g., that the abusive language with time evolves new slurs and clever ways to avoid being detected. Hence they performed a longitudinal study over one year to see how trained models react over time, employing n-grams, word embeddings, and other linguistic and syntactic features. All features combined yielded the best performing model; however, looking at individual features, character n-grams performed best, a result that also Waseem and Hovy (2016) reported.

Transferring knowledge from word embeddings to be used as input to neural networks has been a common technique. Gambäck and Sikdar (2017) experimented with character n-grams in combination with word embeddings from word2vec Mikolov et al. (2013) in various Convolutional Neural Network (CNN) setups, with the best performing model using transferred knowledge from word2vec. Adding character n-grams boosted precision, but lowered recall. Badjatiya et al. (2017) experimented with several machine learners and neural networks, with the best performer being an Long Short-Term Memory (LSTM) with random word vectors where the network's output was used as input to a Gradient Boosted Decision Tree. However, their results have shown questionable and difficult to reproduce (Mishra et al., 2018; Fortuna et al., 2019). Pavlopoulos et al. (2017a,b) tested word embeddings from both GloVe and word2vec in an Recurrent Neural Network (RNN), while Pitsilis et al. (2018) utilised an RNN ensemble, although without use of word embeddings, but feeding standard vectorized word uni-grams to multiple LSTM networks, aggregating the classifications, to outperform the previous state-of-the-art.

Park and Fung (2017) created a hybrid system that tried to capture features from two input levels, using two CNNs, one character-based and one word-based. Meyer and Gambäck (2019) proposed an optimised architecture combining components with CNNs and LSTMs into one system. One part of the system used character n-grams as input while the other part used word embeddings. They used the dataset from Waseem and Hovy (2016), obtaining better results than previous solutions. Most of the research discussed above used that dataset (with labels 'Sexist', 'Racist' or 'Neither') or a slightly modified version (Waseem, 2016).

The dataset by Davidson et al. (2017) in contrast separates hateful language from offensive and normal language, making the task harder. Zhang et al. (2018) used this dataset and six other, but merged the offensive class with the normal class. On the 2-class hate vs normal language task, they outperformed the state-of-the-art on 6 out of 7 datasets with a system feeding word embeddings from word2vec into a CNN to produce input vectors for an LSTM network with GRU cells performing the final classification. Founta et al. (2018a) used the same dataset, but kept the offensive samples separate from the normal ones, thus taking on the challenge of separating hateful and offensive language. They ran two networks in parallel, one RNN with text input and one feed-forward network with metadata input, followed by a concatenation layer and a classification layer, performing slightly below the F_1-score 0.900 Davidson et al. (2017) achieved with a baseline LR model. However, Kshirsagar et al. (2018) surpassed the baseline using pre-trained word embeddings as input to multiple Multilayer Perceptron (MLP) layers, achieving a total F_1-score of 0.924. Still, the F-score increase is due to better performance on the 'Normal' and 'Offensive' classes, with the model actually performing worse on the 'Hate' class.

This agrees with Malmasi and Zampieri (2018) who tested several supervised learners and ensemble classifiers on the dataset, reporting a noticeable difficulty of distinguishing hateful language from profanity. Their extensive results analysis showed that tweets with the highest probability of being tagged as hate usually are targeted at a specific social group, so that contextual and semantic document features may be required to improve performance. Gaydhani et al. (2018) in contrast claimed near-perfect performance, misclassifying only 0.035% of true hate speech samples on a combination of datasets from Davidson et al. and Waseem (2016) using n-grams as features and feeding the TF-IDF values of these into classifiers such as Support Vector Machine, Naïve Bayes and Logistic Regression. However, analysing their training and test data[3] shows that 74% of the test data is either duplicate or in the training data, giving a highly biased test set and questionable results.

Basile et al. (2019) and Zampieri et al. (2019, 2020) present findings from SemEval-2019 Task 5 and 6 resp. -2020 Task 12, observing that pre-trained attention-based deep learning models were used by the top teams in all subtasks. Pérez and Luque (2019) and Indurthi et al. (2019) were the top teams in SemEval-2019 Task 5, using ELMo together with LSTM networks. ELMo (Embeddings from Language Model; Peters et al., 2018) uses a bidirectional Language Model to create deeply contextualised word representations, with unsupervised pre-training. GPT (Generative Pre-training Transformer; Radford et al., 2018, 2019) expanded the amount of text the language model can be trained on by combining the ideas of unsupervised pre-training (Dai and Le, 2015) and transformers (Vaswani et al., 2017) with attention. BERT (Devlin et al., 2019) is a direct descendant of GPT, although instead of using a stack of transformer decoders, BERT uses a stack of transformer encoders, and while GPT only trains a forward language model, BERT is bidirectional. With the release of two pre-trained language models, $BERT_{BASE}$ and $BERT_{LARGE}$, BERT can be used as a language model for tasks such as hate speech detection. Liu et al. (2019) used $BERT_{BASE}$ to deliver some of the best results in SemEval-2019 Task 6, while several SemEval-2020 tasks saw continuous transformer multitask pre-training (ERNIE 2.0; Sun et al., 2020) outperforming other solutions.

4 Architecture

Word embedding techniques based on bag-of-words contexts, such as word2vec (Mikolov et al., 2013), only capture the semantic relations among words (Vashishth et al., 2019), whereas language models are more complex and can capture the meaning of a word in a sentence, i.e., its context. This work focuses on such language models and explores the effect of transferring knowledge from a substantial pre-trained language model to a classifier predicting hateful and offensive expressions.

4.1 Preprocessing

Twitter authors often make use of abbreviations and internet slang. Many tweets in addition contain retweeted content, mentions of other users, URLs, hashtags, emojis, etc. As language models can capture context between words and prefer complete sentences, only simple preprocessing was used to clean the data. NLTK's (Bird et al., 2009) `TweetTokenizer` was used to remove URLs, numbers, mentions and 'RT' retweet marks. Stop words were *not* removed to keep as much context as possible for the language model to capture.

HuggingFace's `BertTokenizer` was used for text normalisation and punctuation splitting

[3] https://github.com/adityagaydhani14/Toxic-Language-Detection-in-Online-Content

as well as WordPiece subword-level tokenisation. Words that do not occur in the vocabulary are segmented into subword units, so there are no out-of-vocabulary words.

4.2 BERT Model Architecture

BERT's language models can be pre-trained from scratch using only a plain text corpus or fine-tuned with a domain-specific corpus. Although pre-training is a one-time procedure, it is relatively expensive requiring a large amount of crawled text and computational power. However, Devlin et al. (2019) released several pre-trained models, two of which were used in the experiments: **BERT Base**, Uncased (12 encoder layers with 768 hidden units and 12 attention heads; 110M parameters) and **BERT Large**, Uncased (24-layer, 1024-hidden, 16-heads; 340M parameters), that were trained on the English Wikipedia and BookCorpus (Zhu et al., 2015) for 1M update steps. Both models are lowercased and have pre-trained checkpoints that can either be trained with more data or fine-tuned with task-specific data. Both of these approaches were implemented and tested in the experiments. The models are trained with word sequence length up to 512, but this can be shorted when fine-tuning, to save substantial memory. Each encoder in the stack applies self-attention and then passes the results through a simple feed-forward network, before handing the output over to the next encoder.

Most language models pass each input token through a token embedding layer to achieve a numerical representation. BERT solves this by passing each token through three different embedding layers (token, segment and positional). Each of these three layers converts an input sequence of tokens to a vector representation of size $(n\,, 768)$, where n is the number of tokens in the input sequence. These three vector representations are summed element-wise to construct a single vector used as input for BERT's encoder stack.

The model output is where BERT separates itself from a traditional transformer: Each token position in the input sequence outputs a length 768 hidden vector for BERT Base and 1024 for BERT Large. Each encoder outputs hidden vectors that can be used as contextualised word embeddings that can be fed into an existing model. For the fine-tuning approach, only the hidden vectors from the final encoder in the stack are relevant and only the hidden vector in the first position is used for sentence classification. This vector can be used as input to any classifier. Devlin et al. achieved great results using only a single-layer network, but the final systems used here are slightly modified with an additional linear layer of size 2048 added to increase the complexity of the model. (An RNN model was also tested, but omitted as learning did not improve.)

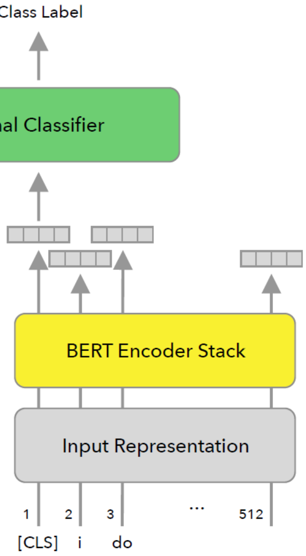

Figure 1: System architecture. The final classifier includes a feedforward network with one input layer, one hidden layer, one output layer, and one softmax layer.

For fine-tuning, only the number of labels needs to be added as a new parameter, 3 and 4 for the systems used here. All BERT parameters and the final classifier network parameters are fine-tuned jointly to maximise the systems' predictive capabilities. The logits from the last linear layer are passed through a softmax layer to calculate the final label probabilities. Between BERT's pooled output and the first linear layer, and between the first and second linear layers, dropout is utilised to regulate the systems to reduce the risk of overfitting. In addition, cross entropy is used to calculate the classification error of each sample. To update the whole network's weights iteratively based on the training data, HuggingFace's version of the Adam optimiser (Kingma and Ba, 2017) is used with weight decay fix, warmup, and linear learning rate decay. Figure 1 gives an overview of the system architecture implemented for the experiments.

4.3 Further Language Model Training

Starting from BERT's Wikipedia and BookCorpus checkpoint, it is possible to further train the language model with domain-specific corpora. This technique of using unlabelled data from the same domain as the target task to train the language model further using the original pre-training objective(s) was first seen in ULMFiT (Howard and Ruder, 2018). Since the approach taken here only uses two datasets, there are still a lot of datasets from the target domain available. Remember, the pre-training only requires the raw text, and so the labels are irrelevant. All available datasets mentioned at the beginning of Section 2, except the two used for the target task, were collected and used to further train BERT on domain data. Furthermore, BERT's English vocabulary consists of 30,522 segmented subword units learned beforehand. Some vocabulary entries are placeholders that can be replaced with new words. ElSherief et al. (2018) created a list of keywords commonly used as hate speech, and most of those were placed in the unused placeholders when further training BERT from its checkpoints.

One of BERT's pre-training objectives is next sentence prediction in which the model predicts whether one sentence follows another sentence or not. As a result, the input format for further training BERT is a single file with untokenised text and one sentence per line. Natural Language Toolkit (NLTK)'s `sent_tokenizer` was used to split documents into sentences of at least one word. Since tweets rarely consist of multiple complete sentences due to Twitter's 280 character limit, some tweets were split in the middle to construct two sentences instead of discarding them.

Other datasets were formatted more easily, e.g, the Stormfront forum data from de Gibert et al. (2018) contained a large folder where each text file was a sentence. All text data from the datasets were merged into one file yielding one large text file with nearly 170,000 lines. This file was then used to further train two language models from BERT Base and Large checkpoints on the two original pre-training objectives, masked LM and next sentence prediction. The output of this process, two language models, trained on Wikipedia, BookCorpus, and domain data was used in the experiments to investigate the effect of further training the language model with domain-specific data.

5 Experiments and Results

The two original pre-trained language models BERT Base and BERT Large from Devlin et al. (2019) were tested together with the two language models (BERT Base* and BERT Large*) further trained with domain-specific data. Each system's performance was tested with the two datasets **D** (Davidson et al., 2017) and **F** (Founta et al., 2018b). Dataset **F** annotates tweets as 'Hateful', 'Offensive' 'Spam' or 'Normal'. When identifying hateful and offensive language, the 'Spam' class is redundant and was omitted. However, to compare to previous research, experiments with the original 4-class dataset **F** were also carried out.

All text data in the experiments were lowercased. Both datasets were split into a training set containing 80% of the total samples and a held-out test set containing the remaining 20%, with Scikit-learn's stratified splitting function used to ensure equal class balance between the sets. The order of the training samples was shuffled before each run. Cross-validation with multiple folds was not implemented due to framework limitations.

All experiments were run on devices with at least 64GB RAM, the amount recommended by the creators of BERT. The two original language models were pre-trained with a sequence length of 512 and batch size 256. The fine-tuned models had a sequence length of 128 and batch size 32. All four language models were trained with the Adam optimiser, with the optimal learning rates found to be 3e-5 for the fine-tuning process and 2e-5 for the classification process after an exhaustive search with parameters suggested by Devlin et al. (2019). Other parameters shared by the four systems are a dropout probability of 10% on all layers, the number of training epochs which was 3, and an evaluation batch size of 8. The fine-tuning of the language models took around 3 hours on two Nvidia V100 GPUs with 32GB RAM each, while classification with BERT Base and Large took on average around 1 and 2 hours, respectively.

System performance will be measured by *micro* averaged Precision, Recall, and F_1-score, as this is more suitable for unbalanced datasets and gives detailed insights into how the models classify each sample. The *macro* averaged total for each metric will also be presented for comparison reasons.

5.1 Dataset from Davidson et al. (2017)

Dataset **D** is quite unbalanced with 77% of the tweets being annotated as 'Offensive' and only 6% being labelled 'Hateful'. As seen in Table 3, all four models perform more or less equally in almost all metrics, and are able to correctly classify tweets as 'Normal' and 'Offensive' fairly well. BERT

BERT Model:		Base	Large	Base*	Large*
Normal	P	0.867	**0.889**	0.883	0.883
	R	**0.906**	0.888	0.893	0.888
	F_1	0.886	**0.888**	**0.888**	0.885
Offensive	P	**0.941**	0.938	0.929	0.932
	R	0.953	0.959	**0.965**	0.961
	F_1	0.947	**0.948**	0.947	0.946
Hateful	P	0.497	**0.520**	0.477	0.460
	R	0.343	**0.364**	0.213	0.259
	F_1	0.406	**0.428**	0.294	0.331
Micro avg.	F_1	0.910	**0.913**	0.909	0.908
Macro avg.	F_1	0.751	**0.759**	0.725	0.729

Table 3: Results from BERT experiments on dataset **D**

BERT Model:		Base	Large	Base*	Large*
Normal	P	0.956	0.956	0.956	**0.957**
	R	**0.968**	0.967	0.967	0.964
	F_1	**0.962**	0.961	0.961	0.960
Offensive	P	**0.865**	0.861	0.860	**0.865**
	R	0.920	**0.926**	0.921	0.919
	F_1	**0.892**	**0.892**	0.889	0.891
Hateful	P	0.573	**0.574**	0.531	0.485
	R	**0.299**	0.264	0.264	0.284
	F_1	**0.393**	0.362	0.353	0.358
Micro avg.	F_1	**0.923**	0.922	0.921	0.919
Macro avg.	F_1	**0.762**	0.756	0.748	0.745

Table 4: Results from BERT experiments on dataset **F**

Large is the model that performs best with a final macro averaged F_1-score of 0.759, with the other three models not far behind. This is in line with Devlin et al. (2019) who found BERT Large outperforming BERT Base across all tasks tested.

Out of the four models, BERT Large also obtains the best scores for the 'Hateful' class, with precision, recall and F_1-score of 0.520, 0.364 and 0.428, respectively. 52% of the examples the model predicted as hateful were correctly classified. Only 36% of the total true hateful tweets were classified correctly, yielding low recall. The two models with general language understanding, BERT Base and Large, outperform the two models with domain-specific language understanding on the 'Hateful' class. On this class, BERT Large* obtains a F_1-score of 0.331 compared to BERT Large's F_1-score of 0.428. This gap in F_1-scores is unexpected as the intention of further training the language models with domain-specific data was to increase the hateful language understanding.

5.2 Dataset from Founta et al. (2018b)

Dataset **F** is nearly three times the size of **D**. The label distribution is also more balanced with roughly half of the samples labelled 'Normal' and the rest distributed between the other three classes. Although only 6% of the tweets are annotated 'Hateful', this is a fair representation of the real world where only a small portion of the online content is hate speech. The best scores for each metric were then spread across the four models and there was no clear difference between the models: all obtained an F_1-score of 0.67. As with dataset **D**, the models were able to correctly classify tweets as 'Normal' and 'Offensive' quite well while misclassifying most of the true 'Hateful' and 'Spam' tweets. The best F_1-scores for the 'Normal' and 'Offensive' classes were 0.869 and 0.884, respectively, obtained by BERT Base*, but the other models were right behind. The only telling difference between the models was the scores on the 'Hateful' class, with BERT Base the clear winner.

Removing the 'Spam' class from the original dataset, we immediately see an increase in the models' scores for all three classes as shown in Table 4. As expected, the increase is most noticeable for the 'Normal' class which previously was highly confused with the 'Spam' class. The increase is less notable for the 'Hateful' class although BERT Base outperforms the other models by a margin. BERT Base is surprisingly the model that performs best overall, beating the other three models on nearly every metric. Remarkably 97% of the tweets labelled as 'Normal' are correctly classified by the model, but only 30% of true hateful samples. Again, the models seem to recognise true hate speech as less hateful than the annotators. The two models trained with domain-specific data, BERT Base* and BERT Large*, perform worse on the 'Hateful' class than the other two models. This is an interesting observation as more training with domain-specific data has shown to increase the performance of models in previous solutions.

6 Evaluation and Discussion

The main difference between the two datasets used in the experiments is the size and label distribution. The size of dataset **F** allows for more training samples than dataset **D** although systems transferring knowledge from pre-trained language models have shown that even small datasets can achieve similar performance (Howard and Ruder, 2018). The four models' overall performance on datasets **D** and **F** are the same despite the fact that the latter dataset allows for more language model fine-tuning. The label distribution in dataset **F** is more realistic than dataset **D**, where a large portion of the samples is labelled as 'Offensive'. However, this unbalance of dataset **D** does not seem to affect the models' performance noticeably. The reason is probably that dataset **D** contains a sufficient amount of class samples for the models to learn the other two classes. This ability to learn with a few training examples is one of the main advantages of using language models instead of traditional word embeddings.

6.1 Language Model Selection

Although datasets without the distinction between offensive and hateful language were irrelevant for testing the models in the experiments, they were used as unlabelled data to further pre-train two BERT language models. This additional training is intended to give the language models domain-specific language understanding and has shown to increase the overall performance in other tasks (Devlin et al., 2019). However, the results obtained from the experiments show that the two models with domain-specific language understanding performed worse or equal to the language models with general language understanding. As we can see in Table 3, the worst performance of the two extended BERT models was on dataset **D**. BERT Base* and Large* obtained macro-averaged F_1-scores of 0.725 and 0.729, respectively, while the original BERT models obtained F_1-scores of 0.751 for BERT Base and 0.759 for BERT Large. The difference between these scores is a result of the models' performance on the 'Hateful' class as the performance on the 'Normal' and 'Offensive' classes are near identical for all four models. BERT Large outperforms the other three models on the 'Hateful' class with a F_1-score of 0.428. This is in line with Devlin et al. (2019) who found that BERT Large outperformed BERT Base on several other tasks.

However, this is not the case for the results obtained by BERT Large on dataset **F**. Looking at Table 4, we observe that the smaller model BERT Base outperforms BERT Large on nearly every metric. The most compelling difference can again be seen in the 'Hateful' row, where BERT Base achieved an F_1-score of 0.393 compared to BERT Large's F_1-score of 0.362, mainly as a result of better recall obtained by BERT Base.

Surprisingly, there is no telling difference when comparing the two models with general language understanding to the two models with domain-specific language understanding. Further training with large domain-specific corpora is expected to be beneficial and increase the performance on downstream tasks like hate speech detection. However, the results from the experiments do not reflect this assumption, and it seems like all four models are able to capture similar features, thus performing equally well. Next sentence prediction is one of BERT's two pre-training objectives. So in order to further pre-train the language model, it is necessary to obtain documents containing at least two sentences. This became a limitation, as the domain-specific data used in the experiments mostly consist of tweets, that often contain only a single sentence and omitting every single-sentence tweet would lead to a much smaller training corpus. In order to include single-sentence tweets in the training corpus, they were split at the middle. This is not optimal and may be one of the reasons why BERT Base* and Large* did not perform as expected.

6.2 Error Analysis

Generally, the results from each dataset indicate that it is hard to separate hateful language from offensive and normal language. This was also the key finding stated by Malmasi and Zampieri (2018) and Davidson et al. (2017) when testing their models' performance on dataset **D**. For dataset **D**, most of the annotated hateful samples are confused with the 'Offensive' class, and this may be due to the skewed dataset where the 'Offensive' samples dominate. With dataset **F**, there is roughly an equal distribution of misclassifications between the 'Offensive' and 'Normal' class. This indicates that neither of the tested models using features from the pre-trained language model is capable of distinguishing hateful language from offensive and neutral language with acceptable accuracy.

To investigate BERT Base's predictions on dataset **F** deeper, some correctly and incorrectly classified instances were sampled and analysed. The model tends to predict instances containing clear racist or homophobic slurs as hate speech, while obvious hate speech appears more straightforward for the model to understand and accurately predict. Several instances annotated as 'Hateful', but predicted as 'Normal' or Offensive' by the model do not appear to be clear hate speech and are perhaps mislabelled by the human coders and

System	P	R	F_1
BERT Large	**0.91**	**0.91**	0.90
Davidson et al. (2017)	**0.91**	0.90	0.90
Founta et al. (2018a)	0.89	0.89	0.89
Kshirsagar et al. (2018)	–	–	**0.92**

Table 5: Dataset **D** comparison (weighted averages)

System	P	R	F_1
BERT Base*	0.800	0.812	**0.806**
BERT Large*	0.802	0.809	0.805
Lee et al. (2018) CNN$_w$	0.789	0.808	0.783
Lee et al. (2018) RNN-LTC$_w$	**0.804**	**0.815**	0.805

Table 6: Dataset **F** comparison (weighted averages)

System	P	R	F_1
BERT Base	0.80	**0.73**	**0.76**
Naïve Bayes	0.63	0.63	0.63
Support Vector Machine	**0.87**	0.65	0.74
Logistic Regression	0.80	0.69	0.74

Table 7: Dataset **F** without "Spam" (macro averages)

correctly predicted by the model. The text "ISIS message calls Trump 'foolish idiot'" was found four times in the original dataset with different authors, being annotated twice as 'Hateful' and twice as 'Offensive', with the model predicting the human-chosen label on only one of the instances. As stated by Chatzakou et al. (2017), annotation is even hard for humans and this is an example of the gold standard not being perfect even though the Founta et al. dataset was thoroughly constructed.

6.3 Comparison to State-of-the-Art

Table 5 shows the results obtained on dataset **D** by BERT Large compared to previous results. Although the dataset is widely used, some researchers (e.g., Zhang et al., 2018) chose to merge the 'Offensive' and 'Normal' classes into one non-hate class; making them not comparable to the results carried out in the experiments. All four systems in Table 5 perform equally well with F_1-scores around 0.90. BERT Large is outperformed by Kshirsagar et al. (2018)'s Transformed Word Embedding Model (TWEM). BERT Large outperforms the solution from Founta et al. (2018a) and obtains similar results as the baseline from Davidson et al. (2017).

Lee et al. (2018) tested several machine learning algorithms on dataset **F** intending to create a baseline for this dataset. Table 6 shows that the two BERT models and Lee et al.'s word-based RNN-LTC model perform similarly on this dataset. However, BERT Base* achieves an F_1-score of 0.361 on the 'Hateful' class, compared to the RNN-LTC model's F_1-score of 0.302. This indicates that BERT Base* is better at separating hateful language from the other types of language. RNN-LTC outperformed BERT Base* on the 'Spam' class resulting in the similar total average scores.

The experimental results on dataset **F** without the 'Spam' class were compared to three baseline systems, since no comparable research was found. The macro-averaged scores are shown in Table 7. Out of the four tested models, BERT Base was the best performing with an F_1-score of 0.76. Again, BERT Base's performance on the "Hateful" class is compellingly better than the best performing Logistic Regression model. BERT Base obtain an F_1-score of 0.393 while the LR model achieves an F_1-score of 0.310. The improved performance on the "Hateful" class on both version of dataset **F** implies that models transferring knowledge from pre-trained language models are able to distinguish the nuances of abusive language more accurately.

Model selection is important when creating a hate speech predictor; however, Gröndahl et al. (2018) argue that model architecture is less important than the type of data and labelling criteria. They found that the tested models, which ranged from simple Logistic Regression to more complex LSTM, performed equally well when recreating several state-of-the-art solutions. Gröndahl et al.'s results are consistent with the investigations conducted during the experiments, where changes in the final classifier's complexity did not reflect any changes in the results.

7 Conclusion and Future Work

To explore the effects of applying language models to the downstream task of hate speech detection, four systems based on the BERT language models were implemented and tested on two datasets annotated both for hateful and offensive language. Two of the systems were further pre-trained with unlabelled domain-specific data. However, the results did not reflect any notable improvement with the extended language models.

All four models achieved F_1-scores close to or above state-of-the-art solutions on both datasets, and their ability to correctly distinguish hate speech from offensive and ordinary language was considerably better than the compared solutions, but the scores on the 'Hateful' class are not sufficient enough to bring the systems into practical use, as hateful expressions would pass through the system or more benign cases would be incorrectly censored. Still, language models bring a considerable potential to understanding all the nuances of hateful utterances, and further exploration of how to most effectively train and transfer knowledge from them is necessary.

The models used in the experiments were all pre-trained on the English Wikipedia and BookCorpus to obtain general language understanding. Typically, the language that appears in Wikipedia articles and books are somewhat domain neutral and formal. This language may be too different from the hate speech domain in terms of words and sentences. Therefore, it may be beneficial to collect documents from hate speech datasets and create one large corpus, which can be used as input data to pre-train BERT's encoders from scratch.

A problem with BERT is the vast number of parameters that need to be set, leading to memory problems and long training times. However, the usage of transformers for language processing is a fast-moving field, so several ideas and strategies have lately been introduced to improve on the original BERT setup. One of those — such as ALBERT, 'A lite BERT' (Lan et al., 2020); GPT-3, 'Generative Pre-trained Transformer' (Brown et al., 2020); continuous pre-training ('ERNIE 2.0'; Sun et al., 2020); transformers for longer sequences ('BigBird'; Zaheer et al., 2020); or layerwise adaptive large batch optimisation ('LAMB'; You et al., 2020) — could be tested on the task.

Lan et al. (2020)'s ALBERT can drastically reduce the number of parameters and help solve memory problems and reduce training times. Zaheer et al. (2020)'s 'BigBird', with its sparse attention mechanism, allows for longer input sequences than BERT and is suitable for tasks where the datasets include longer documents. You et al. (2020) utilised large batch stochastic optimisation methods to reduce the training time of BERT remarkably.

As describe in Section 4.3, each tweet in the training set was split into two for the next sentence prediction task BERT is performing during pre-training. This was done because tweets rarely contain two full sentences. However, this strategy can lead to some loss of linguistic information and it may be better to just skip next sentence prediction during training and only perform the masked language model task.

References

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

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. 2017. Mean birds: Detecting aggression and bullying on Twitter. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 13–22, Troy, New York, USA. Association for Computing Machinery.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, volume 2, pages 3079–3087, Montréal, Québec, Canada. Curran Associates, Inc.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International Conference on Web and Social Media*, pages 512–515, Montréal, Québec, Canada. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Mai ElSherief, Shirin Nilizadeh, Dana Nguyen, Giovanni Vigna, and Elizabeth Belding. 2018. Peer to peer hate: Hate speech instigators and their targets. In *Twelfth International Conference on Web and Social Media*, pages 52–61, Stanford, California, USA. AAAI Press.

Paula Fortuna, Juan Soler-Company, and Sérgio Nunes. 2019. Stop PropagHate at SemEval-2019 tasks 5 and 6: Are abusive language classification results reproducible? In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 745–752, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Antigoni Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. 2018a. A unified deep learning architecture for abuse detection. *CoRR*, abs/1802.00385.

Antigoni Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018b. Large scale crowdsourcing and characterization of Twitter abusive behavior. In *Twelfth International Conference on Web and Social Media*, pages 491–500, Stanford, California, USA. AAAI Press.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Lei Gao and Ruihong Huang. 2017. Detecting online hate speech using context aware models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 260–266, Varna, Bulgaria. INCOMA Ltd.

Aditya Gaydhani, Vikrant Doma, Shrikant Kendre, and Laxmi Bhagwat. 2018. Detecting hate speech and offensive language on Twitter using machine learning: An N-gram and TFIDF based approach. *CoRR*, abs/1809.08651.

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

Jennifer Golbeck, Zahra Ashktorab, Rashad O. Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A. Geller, Quint Gergory, Rajesh Kumar Gnanasekaran, Raja Rajan Gunasekaran, Kelly M. Hoffman, Jenny Hottle, Vichita Jienjitlert, Shivika Khare, Ryan Lau, Marianna J. Martindale, Shalmali Naik, Heather L. Nixon, Piyush Ramachandran, Kristine M. Rogers, Lisa Rogers, Meghna Sardana Sarin, Gaurav Shahane, Jayanee Thanki, Priyanka Vengataraman, Zijian Wan, and Derek Michael Wu. 2017. A large labeled corpus for online harassment research. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 229–233, Troy, New York, USA. Association for Computing Machinery.

Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N. Asokan. 2018. All you need is “love”: Evading hate speech detection. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, AISec ’18, pages 2–12, Toronto, Ontario, Canada. Association for Computing Machinery.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *CoRR*, abs/1801.06146.

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

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Rohan Kshirsagar, Tyrus Cukuvac, Kathy McKeown, and Susan McGregor. 2018. Predictive embeddings for hate speech detection on Twitter. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 26–32, Brussels, Belgium. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations*, Addis Ababa, Ethiopia. OpenReview.net.

Younghun Lee, Seunghyun Yoon, and Kyomin Jung. 2018. Comparative studies of detecting abusive language on twitter. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 101–106, Brussels, Belgium. Association for Computational Linguistics.

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

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

Johannes Skjeggestad Meyer and Björn Gambäck. 2019. A platform agnostic dual-strand hate speech detector. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 146–156, Florence, Italy. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1088–1098, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 145–153, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on Twitter. In *Proceedings of the First Workshop on Abusive Language Online*, pages 41–45, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017a. Deep learning for user comment moderation. In *Proceedings of the First Workshop on Abusive Language Online*, pages 25–35, Vancouver, BC, Canada. Association for Computational Linguistics.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017b. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark. Association for Computational Linguistics.

Juan Manuel Pérez and Franco M. Luque. 2019. Atalaya at SemEval 2019 task 5: Robust embeddings for tweet classification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 64–69, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Georgios K. Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Detecting offensive language in tweets using deep learning. *Applied Intelligence*, 48(12):4730–4742.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. *CoRR*, abs/2004.14454.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2016. Measuring the reliability of hate speech annotations: The case of the European refugee crisis. In *Proceedings of the 3rd Workshop on Natural Language Processing for Computer Mediated Communication*, pages 6–9, Bochum, Germany. Bochumer Linguistische Arbeitsberichte.

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

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *34th AAAI Conference on Artificial Intelligence*, pages 8968–8975, New York, New York, USA. AAAI.

Kevin Systrom. 2017. Keeping Instagram a safe place for self-expression. Instagram.com.

Kevin Systrom. 2018. Protecting our community from bullying comments. Instagram.com.

Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. 2019. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3308–3318, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, Long Beach, California, USA.

Zeerak Waseem. 2016. Are you a racist or am I seeing things? Annotator influence on hate speech detection on Twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 138–142, Austin, Texas, USA. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California, USA. Association for Computational Linguistics.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training BERT in 76 minutes. In *8th International Conference on Learning Representations*, Addis Ababa, Ethiopia. OpenReview.net.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for longer sequences. *CoRR*, abs/2007.14062.

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

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual offensive language identification in social media (OffensEval 2020). *CoRR*, abs/2006.07235.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on Twitter using a convolution-GRU based deep neural network. In *The Semantic Web: 15th European Semantic Web Conference*, pages 745–760, Cham, Switzerland. Springer.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27, Los Alamitos, California, USA. IEEE Computer Society.