Hate Speech Detection in Twitter using Transformer Methods

Raymond T Mutanga[1], Nalindren Naicker[2], Oludayo O Olugbara[3]
ICT and Society Research Group, Department of Information Systems
Durban University of Technology, South Africa
Durban, 4000

***Abstract*—Social media networks such as Twitter are increasingly utilized to propagate hate speech while facilitating mass communication. Recent studies have highlighted a strong correlation between hate speech propagation and hate crimes such as xenophobic attacks. Due to the size of social media and the consequences of hate speech in society, it is essential to develop automated methods for hate speech detection in different social media platforms. Several studies have investigated the application of different machine learning algorithms for hate speech detection. However, the performance of these algorithms is generally hampered by inefficient sequence transduction. The Vanilla recurrent neural networks and recurrent neural networks with attention have been established as state-of-the-art methods for the assignments of sequence modeling and sequence transduction. Unfortunately, these methods suffer from intrinsic problems such as long-term dependency and lack of parallelization. In this study, we investigate a transformer-based method and tested it on a publicly available multiclass hate speech corpus containing 24783 labeled tweets. DistilBERT transformer method was compared against attention-based recurrent neural networks and other transformer baselines for hate speech detection in Twitter documents. The study results show that DistilBERT transformer outperformed the baseline algorithms while allowing parallelization.**

Keywords*—Attention transformer; deep learning; neural network; recurrent network; sequence transduction

I. Introduction

Social media platforms such as Twitter are publicly accessible digital resources for online communication and collaboration. Despite its popularity and convenience, Twitter is increasingly being used to spread hate speech. The level of anonymity granted by Twitter makes it conducive for the dissemination of hateful speech about people. Furthermore, a proportional relationship between hate speech propagation and the occurrence of hate-related crimes is highlighted in other studies [1, 2]. Given the high volume and nature of messages posted on Twitter, it is imperative to develop ways to curb the dissemination of hateful messages.

Currently, social media companies such as Twitter and Facebook employ human annotators to manually delete messages deemed to be hateful [3]. Moreover, users of these platforms are encouraged to flag and report contents they perceive to be inimical to the public. Nevertheless, these methods are labor-intensive and subject to human judgment [4]. The grave consequences of hate speech propagation and inherent limitations of human annotators have necessitated the development of automated hate speech detection methods that use machine learning algorithms. Machine learning algorithms can be classified into two broad categories, which are classical machine learning and deep learning. Both methods have been exploited and tested for hate speech detection in earlier studies.

Classical algorithms depend on feature engineering, a process which is complex and time-consuming. The complexity of the feature engineering process negatively impacts the capture of semantic and syntactic text representations [5]. Deep learning algorithms perform end-to-end training by allowing highly predictive representations to be effectively coded. Deep neural networks such as recurrent neural network (RNN) can preserve sequence information over time, thereby integrating contextual information better in classification tasks [6]. However, their inherently sequential nature prohibits parallelization, thereby increasing processing time. Moreover, RNN suffers from the limitation of long-term dependency, making it less effective as the hiatus between where information appears and the point where the information is required increases. This is particularly important in context-dependent applications such as hate speech detection.

Researchers have created techniques based on recurrent neural networks in conjunction with the attention mechanism to solve some of these problems. Such attention-based recurrent neural networks allow for the modeling of dependencies regardless of distance between the input or output sequences [7, 8]. However, the inclusion of recurrent neural network prohibits parallel processing and negatively impacts processing time. Due to such limitations, some recent works have focused on improving attention mechanisms. Research in this direction has given birth to transformers that perform sequence transduction entirely based on attention [9]. This allows for capturing relevant information that might be contained in every word within a sentence while allowing parallel processing. Consider the following statements for an example. “Foreigners must fall. They are taking our jobs. Some of them are stealing from us”. Current approaches which are mostly based on traditional deep learning algorithms fail to capture that the word “Some” in the third sentence refers to foreigners because of their reliance on past hidden states to capture dependencies with previous words. Transformer algorithms are designed to capture such long-term dependencies using positional embedding to remember word order in sequences [9]. In addition, parallelization enables faster training when compared to traditional deep learning approaches that are based on sequential processing [9].

Consequently, this research seeks to enhance hate speech detection by capturing long-term dependencies using transformer methods while allowing parallel processing. The remainder of this paper is organized as follows. Section II reviews the related works. Section III describes the materials and methods of the study. Experiments and results of experiments are presented in Section IV. Finally, Section V presents the conclusions and future works.

II. Related Work

Supervised machine learning techniques are the predominant approach used for automated hate speech detection [10]. Hate speech detection can be modeled in machine learning as a dichotomous class or multiclass classification problems that can be addressed adequately using either classical learning algorithms or deep learning algorithms. Classical learning algorithms rely on manually engineered features while deep learning algorithms automatically learn features from the input data, instead of adopting handcrafted features [5]. The unstructured nature of human language presents many intrinsic challenges to automated text classification methods. One key challenge faced by existing methods of hate speech detection is the failure to capture long-term dependencies. This leads to loss of contextual information, which is vital for semantic interpretation. Deep learning algorithms, particularly the recurrent neural network (RNN) algorithms, have been the de-facto methods in handling sequence data such as text [11, 12]. However, they have been limited in the length of sequences they can capture [13]. Transformers are a promising way for capturing long-term dependencies in textual data. However, the technical barriers that need to be surmounted o adapt transformers to automated hate speech detection are not trivial.

RNN algorithms such as long short-term memory (LSTM) [14] and gated recurrent units [15] were developed specifically to address the problem of long-term dependencies which other machine learning algorithms suffer from. The Vanilla RNN works by assigning more weights to prior data points of a sequence, making it suitable for classification of textual data in a way that facilitates improved semantic analysis [16]. Nevertheless, RNN is prone to problems of exploding gradient and vanishing gradient during backpropagation training [17].

The LSTM has a chain-like structure of the Vanilla RNN, but further incorporates multiple gates to control the quantity of data that are allowed into every node state. LSTM is especially helpful in minimizing the vanishing gradient problem [18]. In addition, the LSTM preserves long-term dependencies efficaciously compared to the Vanilla RNN [16], thereby allowing the algorithm to capture more context. Despite these benefits, the LSTM cannot capture long term dependencies to arbitrary lengths. Specifically, the performance of the LSTM drops as sequence length increases beyond thirty words [7].

Further research that is aimed at addressing the problem of long-term dependencies has given birth to the attention mechanism [7]. Attention allows modeling of dependencies irrespective of the distance between input and output sequences [8]. Attention mechanisms work on the assumption that every word in each sentence is relevant. This allows for the capture of context that may be necessary when classifying subjective text such as hate speech. Attention-based models have been investigated with success in text-related tasks [19-21]. However, they are used in conjunction with RNNs [9] and therefore are unable to process word sequences in parallel. For a large corpus of text, this may significantly affect the processing time.

Recent adaptations of attention approach have shifted methods progressively from RNNs to self-attention and transformers [22]. Transformer has rapidly become the dominant architecture for natural language processing (NLP), outperforming RNNs in natural language generation and natural language understanding [23]. The transformer architecture scales well with training data and model size while facilitating efficient parallelization and capturing long-range sequence features. In addition, transformers allow transfer learning by fine-tuning large pre-trained language models for downstream NLP tasks with a relatively small number of training examples, resulting in an improved performance regardless of dataset size [24]. This is particularly important when dealing with highly imbalanced datasets with few instances of hate speech.

There are several types of transformer methods that have been investigated with success in NLP research. Bidirectional encoder representations from text (BERT) [22] has surpassed previous performance benchmarks in common NLP tasks [25]. BERT uses vast unlabeled data for creating models whose parameters can be tuned as desired for smaller supervised data to improve performance. The success of BERT has led to the development of several algorithms based on BERT architecture. These algorithms include RoBERTa [26], DistilBERT [27] and XLNET [28]. RoBERTa is an enhancement of BERT, which is trained on a bigger dataset to improve performance while DistilBERT learns a streamlined version of BERT. XLNET is a generalized autoregressive pre-training method that aims to reconstruct the original data from corrupted input.

III. Materials and Methods

In this section, we present materials and methods used in this study, including acquisition and structure of experimental dataset and setup.

A. Experimental Dataset

Multiclass hate speech and offensive (HSO) language dataset was used in this study for model validation. The dataset was developed, first used by authors in [29], and it was distributed through CrowdFlower. This dataset contains 24783 Twitter text messages that have been labeled into one of the following three classes: 'neutral', 'Offensive' and 'Hate' where 77.4% of the messages are labeled as 'neutral', 16.8% as 'Offensive' and 5.8% as 'Hate'. In this paper, the hate speech detection has been solved as a three classes classification problem.

B. Experimental Setup

The proposed methods of this study were implemented using Python programming language. Keras library was used to implement the attention-based LSTM method. The proposed method was implemented using Hugging face transformers class embedded in Python. Experiments were conducted on a computer running Windows 10 operating system with the configuration of Intel(R) Core (TM) i5-8250U CPU @ 1.60GHz (8 CPUs), 1.8GHz, 8 GB RAM and 500 Gigabytes hard disk drive.

C. Preprocessing

Due to the colloquial nature of Twitter messages, Twitter data are highly unstructured and contain a lot of noise that can affect method accuracy. Consequently, it was deemed necessary to preprocess all Tweets to remove less predictive text features. Preprocessing is widely known to improve performance of classification methods [30] while reducing processing time. The labeled dataset was initially processed to normalize Twitter text as follows:

- Removal of the following noise characters i.e. :| : , ; &amp; ! ? \.
- Normalization of hashtags into words, so that for instance, “#’muslimsmustfall’ becomes ‘‘Muslims must ‘fall’. Hashtags are used to prefix tweets but do not give any valuable information. We have developed a python method to normalize hashtags.
- Lowercasing and stemming to reduce word inflexions.
- Removal of tokens that appear in the dataset less than 5 times that is elimination of low frequency terms at a threshold of 5.

D. Proposed Method

Transformers mirror the standard NLP machine learning method pipeline that includes the processing of data, application of a method, and making predictions. This approach was selected because of its inherent self-attention mechanism that allows it to capture long-term dependencies while allowing parallel processing of input features. The capture of long-term dependencies allows the transformer to perform anaphora resolution, which is one of the major challenges in text processing [31]. However, most transformer models require substantial computational resources, thereby limiting their applicability in resource-constrained environments [32]. For example, they cannot run on portable devices. Furthermore, these models are slower at inference times, making them unfeasible for real-time situations. To address these problems, we propose DistilBERT a streamlined version of BERT that uses only half the number of parameters of BERT [27] but retains the performance of BERT in many text processing tasks [33] while making the inference 60% faster than BERT [34]. DistilBERT was created by removing token type embeddings and pooler from the default architecture of BERT [35]. DistilBERT further reduced the number of layers by 50%, thereby significantly reducing the footprint of the model.

In this study, we have used the distilBERT base uncased model with 66 million parameters pretrained on the Toronto Book Corpus and English Wikipedia [27]. The data used in the experiment was first preprocessed using the steps discussed in Section III of this paper. The preprocessed data were then split where a random 80% of instances were allocated for parameter fine-tuning and training, while 20% random instances were allocated for evaluating the performance of the final model. Fig. 1 shows the architecture of the proposed DistilBERT method of this study. The hyperparameter fine tuning component of the architecture is essential and will be explained in the subsequent section.

E. Model Parameters

The hyperparameter tuning is a crucial step when customizing pre-trained models to specific tasks. As shown in Table I, we optimized our method for hate speech detection by altering sequence length, batch size, early stopping patience and number of epochs. The maximum sequence length was set at 280 in line with the Twitter limit of allowable characters. The optimal number of epochs in our experiments was set to 4. We have configured the early stopping patience technique to prevent our method from overfitting. We set the early stopping patience value at 4, therefore, training is terminated if the evaluation loss fails to improve for four successive evaluations. The evaluation batch size defines the number of examples that are processed concurrently during the training session. In our experiments, we set the evaluation batch size to 256 because it was the largest batch size that our processor could handle effectively. The proposed method has been evaluated using five state of the art metrics outlined in Section IV of this paper.

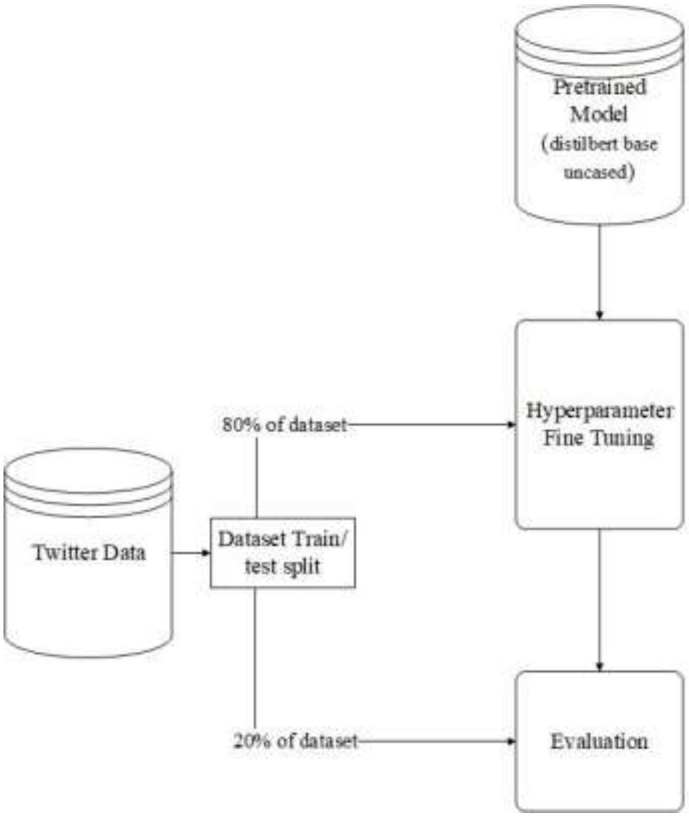
Pretrained Model (distilbert base uncased)
Hyperparameter Fine Tuning
80% of dataset
Twitter Data
Dataset Train/ test split
20% of dataset
Evaluation

Fig. 1. Architecture of the Distilbert Hate Speech Detection Method.

TABLE I. PARAMETERS OF THE PROPOSED APPROACH

Parameter	seq_length	epochs	Early_stopping	Batch_size
Value	280	4	4	256

IV. Results and Discussion

Results of the proposed DistilBERT method was compared against results computed by BERT, XLNet, RoBERTa and attention-based LSTM. We split the dataset in the ratio of 80:20 for model training and testing, respectively. The algorithms were analyzed in terms of six standard functional metrics of accuracy, precision, recall and F-measure, Mathews correlation coefficient (MCC) and evaluation loss. The results are presented based on the ability of the models to detect hate tweets.

A. *Analysis of Accuracy*

The experimental results of the proposed DistilBERT method, along with five baseline algorithms are presented in Table II and Fig. 2. It can be observed that the proposed DistilBERT method recorded the highest average accuracy of 92%. It is worth mentioning that the differences in accuracy scores for all transformer-based methods were negligible. This may be attributed to the fact that they all use standard extensively tested pre-trained models. Expectedly all transformer-based algorithms performed better than the LSTM with Attention. The least performing transformer method had an accuracy of 89%, which is superior to LSTM with attention which had 66% accuracy. This trend is because of the ability of the transformers to capture long-term dependencies better than LSTM with Attention.

B. *Analysis of Precision*

It can be observed from Table III and Fig. 3 that the DistilBERT (base-uncased) and XLNet algorithms jointly recorded the highest precision score of 75% whilst LSTM with attention recorded the least precision score of 65.9%. Although the LSTM with attention recorded the least result, it should be noted that this score is higher than scores recorded by methods using classical machine learning [29]. This result confirms the literature position that attention improves performance in NLP tasks [19].

TABLE II. Accuracy Scores of Four Benchmark Hate Speech Detection Algorithms and the Proposed Distilbert Model on the HSO Dataset

Algorithm	Method Name	Accuracy
BERT	bert-base-uncased	0.90
RoBERTa	robert-base	0.91
RoBERTa	robert-base-openai-detector	0.90
XLNet	xlm-mlm-en-2048	0.91
LSTM with Attention		0.66
DistilBERT	distilbert-base-uncased	0.92

TABLE III. Precision Scores of Four Benchmark Hate Speech Detection Algorithms and the Proposed DistilBERT Model on the HSO Dataset

Algorithm	Method Name	Precision
BERT	bert-base-uncased	0.74
RoBERTa	robert-base	0.74
RoBERTa	robert-base-openai-detector	0.72
XLNet	xlm-mlm-en-2048	0.75
LSTM with Attention		0.66
DistilBERT	distilbert-base-uncased	0.75

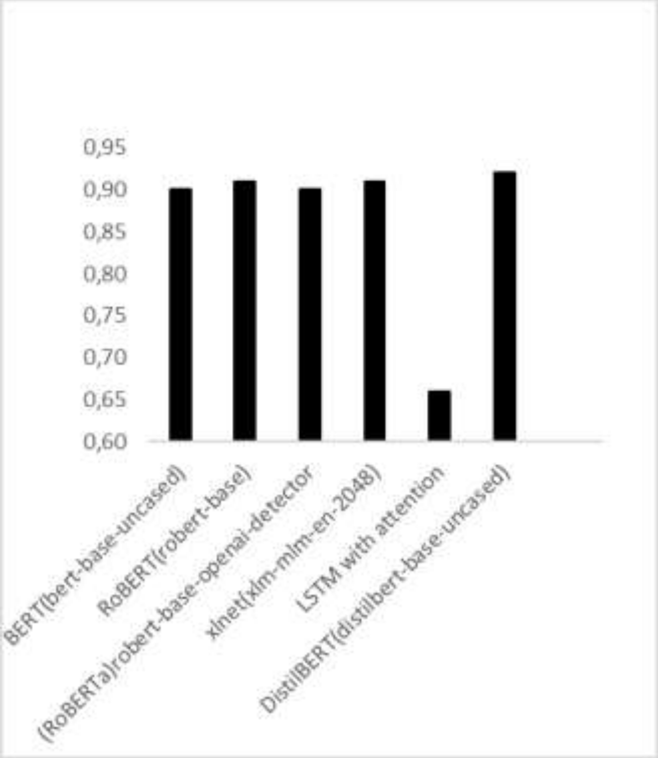

Fig. 2. Illustration of Accuracy Scores using Four Benchmarking Hate Speech Detection Algorithms and the Proposed DistilBERT Model.

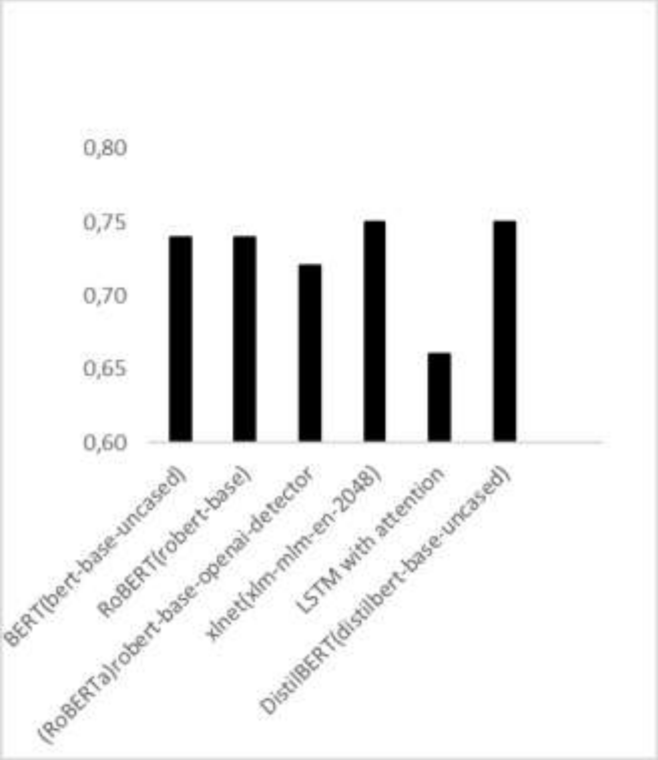

Fig. 3. Illustration of Precision Scores using Four Benchmarking Hate Speech Detection Algorithms and the Proposed DistilBERT Model.

C. Analysis of Recall

Results from Table IV and Fig. 4 show that DistilBERT and XLNet recorded the average recall score of 75% to demonstrate its superior over other algorithms explored in this study. LSTM with attention had the least recall score of 66%. Although the LSTM with attention performed inferior in our experiments, it should be noted that it performed superior to an earlier study on the same dataset for the task of hate speech detection [29].

D. Analysis of MCC Scores

Table I lists the MCC scores calculated for the overall test tweets selected from the experimental dataset. It can be observed that our proposed method recorded the highest MCC score of 75%. Fig. 5 shows that the difference in MCC scores for all algorithms explored in this study is negligible. The worst performing algorithm was RoBERTa (robert-base-openai-detector) which recorded a MCC score of 71% while the best performing algorithm was DistilBERT (distilbert-base-uncased) which recorded a MCC score of 75%.

TABLE IV. RECALL SCORES OF FOUR BENCHMARK HATE SPEECH DETECTION ALGORITHMS AND THE PROPOSED DISTILBERT MODEL ON THE HSO DATASET

Algorithm	Method Name	Recall
BERT	bert-base-uncased	0.72
RoBERTa	robert-base	0.65
RoBERTa	robert-base-openai-detector	0.63
XLNet	xlm-mlm-en-2048	0.69
LSTM with Attention		0.66
DistilBERT	distilbert-base-uncased	0.75

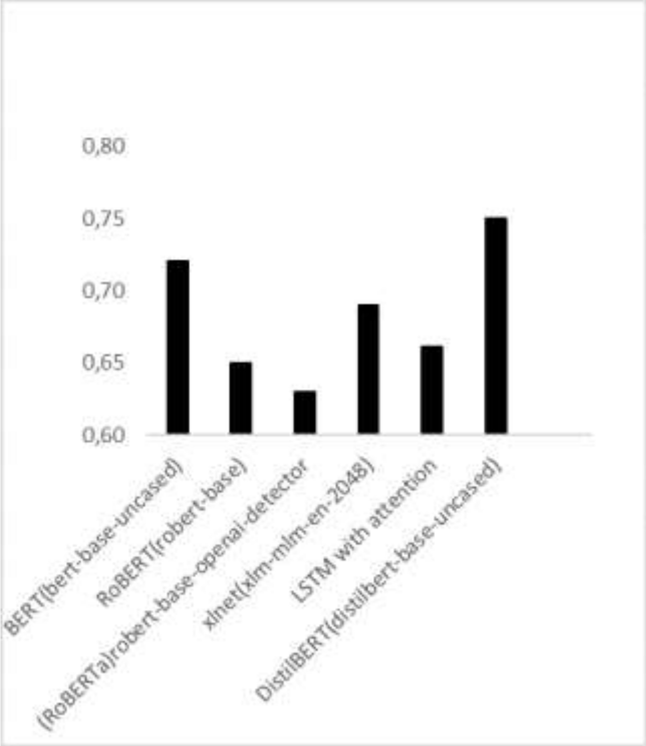

Fig. 4. Illustration of Recall Scores using four Benchmarking Hate Speech Detection Algorithms and the Proposed DistilBERT Model.

TABLE V. MCC SCORES OF FOUR BENCHMARK HATE SPEECH DETECTION ALGORITHMS AND THE PROPOSED DISTILBERT MODEL ON THE HSO DATASET

Algorithm	Method Name	MCC
BERT	bert-base-uncased	0.73
RoBERTa	robert-base	0.73
RoBERTa	robert-base-openai-detector	0.71
XLNet	xlm-mlm-en-2048	0.74
LSTM with Attention		0.72
DistilBERT	distilbert-base-uncased	0.75

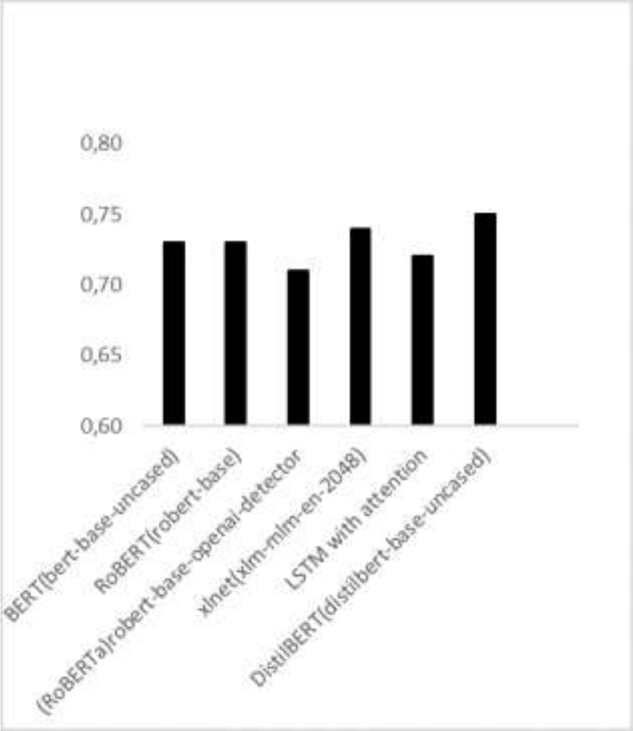

Fig. 5. Illustration of MCC Scores using Four Benchmarking Hate Speech Detection Algorithms and the Proposed DistilBERT Model.

E. Analysis of Evaluation Loss

Table VI shows evaluation loss recordings for the experiments carried out in this study. Fig. 6 clearly shows that our proposed method recorded the best (lowest) evaluation loss of 28% while the LSTM with attention recorded the worst evaluation loss of 36%. This shows that our proposed method maximized predictive capability while minimizing the misclassification error rate more than any of the baseline algorithms.

TABLE VI. EVALUATION LOSS SCORES OF FOUR BENCHMARK HATE SPEECH DETECTION ALGORITHMS AND THE PROPOSED DISTILBERT MODEL ON THE HSO DATASET

Algorithm	Method Name	Eval loss
BERT	bert-base-uncased	0.32
RoBERTa	robert-base	0.32
RoBERTa	robert-base-openai-detector	0.33
XLNet	xlm-mlm-en-2048	0.31
LSTM with Attention		0.36
DistilBERT	distilbert-base-uncased	0.28

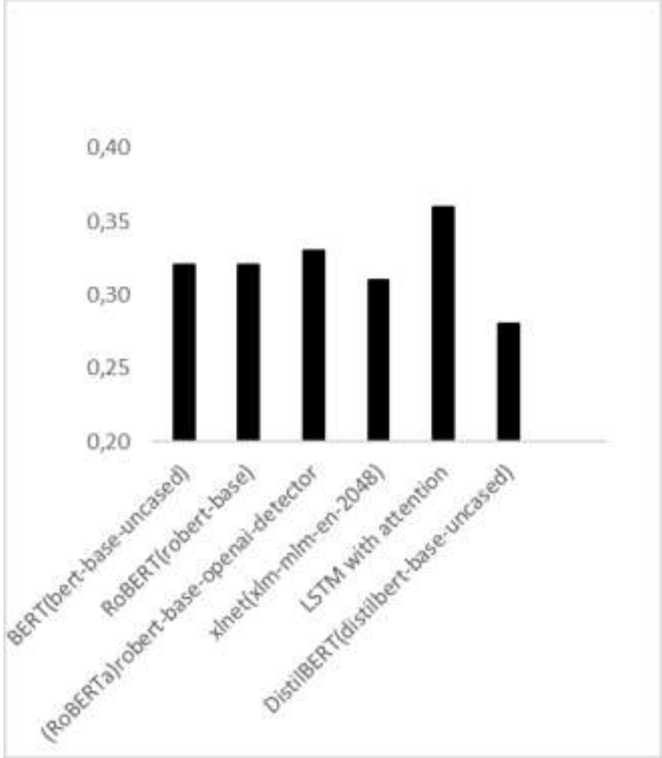

Fig. 6. Illustration of Evaluation Loss Scores using Four Benchmarking Hate Speech Detection Algorithms and the Proposed DistilBERT Model.

F. Analysis of F-Measure

Table VII and Fig. 7 show the F-measure scores of the algorithms explored in this study. It can be observed that the DistilBERT (distilbert-base-uncased) recorded the best F-measure score of 75% while LSTM with attention recorded the lowest F-measure score of 66%. Although DistilBERT has fewer layers and parameters, it outperformed all other transformer algorithms explored in this study. The superior performance of DistilBERT may be attributed to the chosen hyperparameters during experimentation. The same hyperparameters were used to train all the models. It can be argued that the used parameters are not necessarily the optimal combination of hyperparameters for each model explored in this study. Careful selection of the best hypeparameters may improve performance of models such as BERT and RoBERTa.

TABLE VII. F-MEASURE SCORES OF FOUR BENCHMARK HATE SPEECH DETECTION ALGORITHMS AND THE PROPOSED DISTILBERT MODEL ON THE HSO DATASET

Algorithm	Method Name	F-Measure
BERT	bert-base-uncased	0.73
RoBERTa	robert-base	0.69
RoBERTa	robert-base-openai-detector	0.67
XLNet	xlm-mlm-en-2048	0.72
LSTM with Attention		0.66
DistilBERT	distilbert-base-uncased	0.75

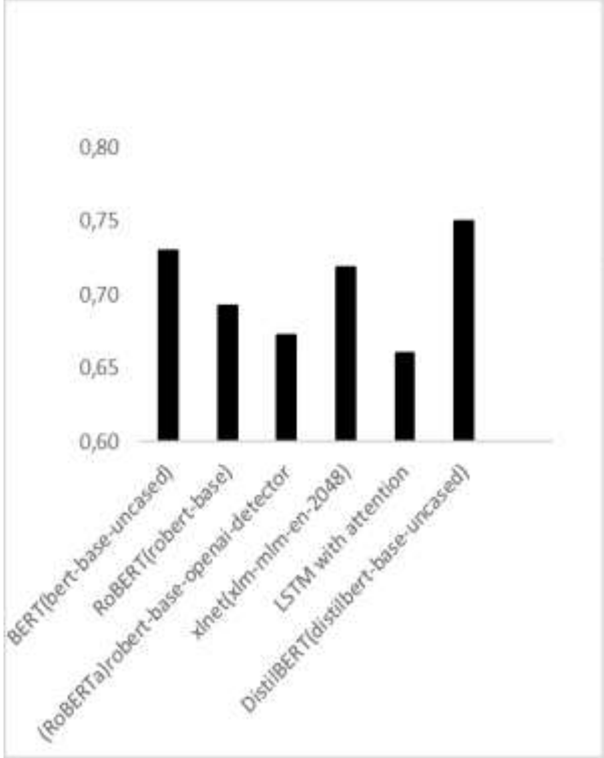

Fig. 7. F-Measure Scores Per Algorithm Illustration of F-Measure Scores using Four Benchmarking Hate Speech Detection Algorithms and the Proposed DistilBERT Model.

Comparative results based on five different metrics from this work show that the transformer models consistently outperform the LSTM with attention. The superior performance of transformer demonstrates that limitations of LSTM, which are inefficient sequence transduction and lengthy processing time have been adequately addressed by the transformer method in hate speech detection.

V. CONCLUSION AND FUTURE WORK

Given the societal implications of hate speech, it is crucial that systems that can accurately distinguish between hate speech, offensive language and neutral speech are developed. Despite concerted efforts from social media companies, governments, and academia, hate speech detection remains a challenging problem in the society of today. In this paper, we have explored several transformer-based methods for hate speech detection. We have evaluated the effectiveness of our method using six state of the art metrics. The results showed that the DistilBERT, a distilled version of BERT, outperforms all transformer-based baseline methods and the attention-based LSTM explored in this study. We, therefore, conclude that the proposed method can be used to learn effective information for the classification of hate speech in resource-constrained environments because it is computationally inexpensive. In addition, transformers facilitate transfer learning, allowing them to be used where training data is limited. It is common for hate speech on social media to be expressed in more than one language. For example, most people in Africa codeswitch their native languages with French, Portuguese, or English language. In future work, we plan to explore multilingual pre-trained models for the task of hate speech detection. The data used in this study were limited to textual Twitter texts only, whereas hate speech on Twitter may be expressed through different data formats such as images and videos. For example, a user may post a video inciting hate speech on Twitter and still go undetected. This limitation calls for the development of multimodal datasets that include other formats of data. Future study will develop methods that integrate both textual and image data for hate speech detection.

ACKNOWLEDGMENTS

We would like to thank the Durban University of Technology for making funding opportunities and materials for experiments available for this research project. Kind acknowledgement to Dr. T. Adeliyi for his contributions.

REFERENCES

[1] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra, "Detection of cyberbullying incidents on the instagram social network," arXiv preprint arXiv:1503.03909, 2015.

[2] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," in European semantic web conference, 2018: Springer, pp. 745-760.

[3] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in Proceedings of the NAACL student research workshop, 2016, pp. 88-93.

[4] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, "Effective hate-speech detection in Twitter data using recurrent neural networks," Applied Intelligence, vol. 48, no. 12, pp. 4730-4742, 2018.

[5] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," ieee Computational intelligenCe magazine, vol. 13, no. 3, pp. 55-75, 2018.

[6] S. Sohangir, D. Wang, A. Pomeranets, and T. M. Khoshgoftaar, "Big Data: Deep Learning for financial sentiment analysis," Journal of Big Data, vol. 5, no. 1, p. 3, 2018.

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

[8] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, "Structured attention networks," arXiv preprint arXiv:1702.00887, 2017.

[9] A. Vaswani et al., "Attention is all you need," in Advances in neural information processing systems, 2017, pp. 5998-6008.

[10] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," in Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, 2017, pp. 1-10.

[11] X. Wang, W. Jiang, and Z. Luo, "Combination of convolutional and recurrent neural network for sentiment analysis of short texts," in Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers, 2016, pp. 2428-2437.

[12] M. S. Islam, S. S. S. Mousumi, S. Abujar, and S. A. Hossain, "Sequence-to-sequence Bangla sentence generation with LSTM recurrent neural networks," Procedia Computer Science, vol. 152, pp. 51-58, 2019.

[13] Z. Mhammedi, A. Hellicar, A. Rahman, and J. Bailey, "Efficient orthogonal parametrisation of recurrent neural networks using householder reflections," in International Conference on Machine Learning, 2017, pp. 2401-2409.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555, 2014.

[16] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," Information, vol. 10, no. 4, p. 150, 2019.

[17] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks, vol. 5, no. 2, pp. 157-166, 1994.

[18] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in International conference on machine learning, 2013, pp. 1310-1318.

[19] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," arXiv preprint arXiv:1509.00685, 2015.

[20] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in Proceedings of the 2016 conference on empirical methods in natural language processing, 2016, pp. 606-615.

[21] B. Yang, L. Wang, D. Wong, L. S. Chao, and Z. Tu, "Convolutional self-attention networks," arXiv preprint arXiv:1904.03107, 2019.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[23] T. Wolf et al., "Transformers: State-of-the-art natural language processing," arXiv preprint arXiv:1910.03771, 2019.

[24] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," arXiv preprint arXiv:2003.00104, 2020.

[25] T. Rajapakse. "To Distil or Not To Distil: BERT, RoBERTa, and XLNet." https://towardsdatascience.com/to-distil-or-not-to-distil-bert-roberta-and-xlnet-c777ad92f8 (accessed 28 July, 2020).

[26] Y. Liu et al., "Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692, 2019.

[27] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.

[28] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in Advances in neural information processing systems, 2019, pp. 5753-5763.

[29] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in Eleventh international aaai conference on web and social media, 2017.

[30] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," Information Processing & Management, vol. 50, no. 1, pp. 104-112, 2014.

[31] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment analysis is a big suitcase," IEEE Intelligent Systems, vol. 32, no. 6, pp. 74-80, 2017.

[32] H. Sajjad, F. Dalvi, N. Durrani, and P. Nakov, "Poor Man's BERT: Smaller and Faster Transformer Models," arXiv preprint arXiv:2004.03844, 2020.

[33] B. Cheang, B. Wei, D. Kogan, H. Qiu, and M. Ahmed, "Language Representation Models for Fine-Grained Sentiment Classification," arXiv preprint arXiv:2005.13619, 2020.

[34] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning based text classification: A comprehensive review," arXiv preprint arXiv:2004.03705, 2020.

[35] B. Büyüköz, A. Hürriyetoğlu, and A. Özgür, "Analyzing ELMo and DistilBERT on Socio-political News Classification," in Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020, 2020, pp. 9-18.