LNCS 13375

Osvaldo Gervasi · Beniamino Murgante ·
Eligius M. T. Hendrix · David Taniar ·
Bernady O. Apduhan (Eds.)

Computational Science and Its Applications – ICCSA 2022

22nd International Conference
Malaga, Spain, July 4–7, 2022
Proceedings, Part I

1 Part I

INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND ITS APPLICATIONS
ICCSA
2022

Springer

Osvaldo Gervasi · Beniamino Murgante ·
Eligius M. T. Hendrix · David Taniar ·
Bernady O. Apduhan (Eds.)

Computational Science and Its Applications – ICCSA 2022

22nd International Conference
Malaga, Spain, July 4–7, 2022
Proceedings, Part I

Springer

Editors
Osvaldo Gervasi
University of Perugia
Perugia, Italy

Eligius M. T. Hendrix
Universidad de Málaga
Malaga, Spain

Bernady O. Apduhan
Kyushu Sangyo University
Fukuoka, Japan

Beniamino Murgante
University of Basilicata
Potenza, Potenza, Italy

David Taniar
Monash University
Clayton, VIC, Australia

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-10521-0 ISBN 978-3-031-10522-7 (eBook)
https://doi.org/10.1007/978-3-031-10522-7

Preface

These two volumes (LNCS 13375–13376) consist of the peer-reviewed papers from the main tracks at the 22nd International Conference on Computational Science and Its Applications (ICCSA 2022), which took place during July 4–7, 2022. The peer-reviewed papers of the workshops are published in a separate set consisting of six volumes (LNCS 13377–13382).

This year, we again decided to organize a hybrid conference, with some of the delegates attending in person and others taking part online. Despite the enormous benefits achieved by the intensive vaccination campaigns in many countries, at the crucial moment of organizing the event, there was no certainty about the evolution of COVID-19. Fortunately, more and more researchers were able to attend the event in person, foreshadowing a slow but gradual exit from the pandemic and the limitations that have weighed so heavily on the lives of all citizens over the past three years.

ICCSA 2022 was another successful event in the International Conference on Computational Science and Its Applications (ICCSA) series. Last year, the conference was held as a hybrid event in Cagliari, Italy, and in 2020 it was organized as virtual event, whilst earlier editions took place in Saint Petersburg, Russia (2019), Melbourne, Australia (2018), Trieste, Italy (2017), Beijing, China (2016), Banff, Canada (2015), Guimaraes, Portugal (2014), Ho Chi Minh City, Vietnam (2013), Salvador, Brazil (2012), Santander, Spain (2011), Fukuoka, Japan (2010), Suwon, South Korea (2009), Perugia, Italy (2008), Kuala Lumpur, Malaysia (2007), Glasgow, UK (2006), Singapore (2005), Assisi, Italy (2004), Montreal, Canada (2003), and (as ICCS) Amsterdam, The Netherlands (2002) and San Francisco, USA (2001).

Computational science is the main pillar of most of the present research, and industrial and commercial applications, and plays a unique role in exploiting ICT innovative technologies. The ICCSA conference series provides a venue to researchers and industry practitioners to discuss new ideas, to share complex problems and their solutions, and to shape new trends in computational science.

Apart from the six main tracks, ICCSA 2022 also included 52 workshops on topics ranging from computational science technologies and application in many fields to specific areas of computational sciences, such as software engineering, security, machine learning and artificial intelligence, and blockchain technologies. For the main conference tracks we accepted 57 papers and 24 short papers out of 279 submissions (an acceptance rate of 29%). For the workshops we accepted 285 papers. We would like to express our appreciation to the workshops chairs and co-chairs for their hard work and dedication.

The success of the ICCSA conference series in general, and of ICCSA 2022 in particular, vitally depends on the support of many people: authors, presenters, participants, keynote speakers, workshop chairs, session chairs, organizing committee members, student volunteers, Program Committee members, advisory committee members, international liaison chairs, reviewers, and others in various roles. We take this opportunity to wholehartedly thank them all.

We also wish to thank our publisher, Springer, for their acceptance to publish the proceedings, for sponsoring some of the best papers awards, and for their kind assistance and cooperation during the editing process.

We cordially invite you to visit the ICCSA website https://iccsa.org where you can find all the relevant information about this interesting and exciting event.

July 2022

Osvaldo Gervasi
Beniamino Murgante
Bernady O. Apduhan

Welcome Message from Organizers

The ICCSA 2021 conference in the Mediterranean city of Cagliari provided us with inspiration to offer the ICCSA 2022 conference in the Mediterranean city of Málaga, Spain. The additional considerations due to the COVID-19 pandemic, which necessitated a hybrid conference, also stimulated the idea to use the School of Informatics of the University of Málaga. It has an open structure where we could take lunch and coffee outdoors and the lecture halls have open windows on two sides providing optimal conditions for meeting more safely.

The school is connected to the center of the old town via a metro system, for which we offered cards to the participants. This provided the opportunity to stay in lodgings in the old town close to the beach because, at the end of the day, that is the place to be to exchange ideas with your fellow scientists. The social program allowed us to enjoy the history of Malaga from its founding by the Phoenicians...

In order to provoke as much scientific interaction as possible we organized online sessions that could easily be followed by all participants from their own devices. We tried to ensure that participants from Asia could participate in morning sessions and those from the Americas in evening sessions. On-site sessions could be followed and debated on-site and discussed online using a chat system. To realize this, we relied on the developed technological infrastructure based on open source software, with the addition of streaming channels on YouTube. The implementation of the software infrastructure and the technical coordination of the volunteers were carried out by Damiano Perri and Marco Simonetti. Nine student volunteers from the universities of Málaga, Minho, Almeria, and Helsinki provided technical support and ensured smooth interaction during the conference.

A big thank you goes to all of the participants willing to exchange their ideas during their daytime. Participants of ICCSA 2022 came from 58 countries scattered over many time zones of the globe. Very interesting keynote talks were provided by well-known international scientists who provided us with more ideas to reflect upon, and we are grateful for their insights.

Eligius M. T. Hendrix

Exploring Neural Embeddings and Transformers for Isolation of Offensive and Hate Speech in South African Social Media Space

Oluwafemi Oriola[1,2](✉) and Eduan Kotzé[1]

[1] University of the Free State, Bloemfontein, South Africa
[2] Adekunle Ajasin University, Akungba, Nigeria
oluwafemi.oriola@aaua.edu.ng

Abstract. Our previous study on classification and detection of abusive language in South African social media space has shown the high prospect of surface level features and classical machine learning models in terms of accuracy. However, much improvement is still needed in the aspect of F1-score. Therefore, the state-of-the-arts such as neural embeddings (Word2Vec, Doc2Vec, GloVe) and neural network-based transformer models (BERT and mBERT) which have performed well in many hate speech isolation tasks are explored in this work. In the evaluation of classical machine learning algorithms, Word2Vec with Support Vector Machine outperformed the previous models, Doc2Vec and GloVe in terms of F1-score. In the evaluation of neural networks, all the neural embedding and transformer models performed worse than the previous models in terms of F1-score. In conclusion, the impressive performance of Word2Vec neural embedding with classical machine learning algorithms in terms of best F1-score of 0.62 and accuracy of 0.86 shows its good prospect in the isolation of abusive language and hate speech in South African social media space.

Keywords: Computational model · Neural models · South African abusive languages · Classification

1 Introduction

Toxic comments are not limited to verbal violence but include harassment, profanity, hate speech and cyberbullying [1], which unfortunately have pervaded the social media landscape. In response to the menace, many research efforts have gone into ridding out these ills from the social media using text mining. In South Africa particularly, they are treated with the acts of parliament using the instrument of laws [2]. However, the dispensation of justice has been impeded by lack of adequate provision of laws or skills, which automatic hate speech isolation tool can ameliorate [3].

According to South African Hate Crime Bill [2], hate speech is hinged on discrimination based on race, colour, ethnicity, gender, sexual orientation, nationality, or religion and targets citizens in manners that are potentially harmful to them. It is different from offensive speech, which is often less discriminatory compared to hate speech [4]. Thus, social media hate speech would be better distinguished from offensive speech and free speech using advanced technology such as Machine Learning and Deep Learning that could carry out in-depth data analysis.

The survey of existing works in the domain of offensive and hate speech isolation according to Fortuna and Nunes [5] showed that surface-level features such as N-Gram, term frequency inverse document frequency (TF-IDF), stylometry and user profile as well as neural embedding and transformer models have been employed during feature extraction and engineering step. Also, classical machine learning algorithms such as Logistic Regression (LogReg), Support Vector Machine (SVM), Random Forest (RF) and Gradient Boosting (GB), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been employed in the classification step, with some impressive outcomes according to Parihar *et al.* [6]. Thus, our initial efforts in this domain went into exploring the surface level features and classical machine learning techniques to segregate offensive and hate speech from free speech in South African context [7]. However, the results obtained are below the expectation particularly in the aspect of F1-score. This paper therefore moves further by exploring neural embedding and transformer models for offensive and hate speech isolation on social media.

2 Background

Since the current work is building on our previous work [7], the subsections below illustrate the key aspects of the previous work.

The previous work focused on automatic detection of hate and offensive South African tweets in a multiclass classification problem space using both classical and ensemble machine learning models. N-gram, syntactic, negative sentiment-based surface level features and their combinations were investigated. Since no corpus existed, we first developed a golden standard Twitter corpus for the South African abusive language context using Twitter Archiver, a Google Sheets plugin which is based on Twitter Search API. We relied on South African laws in [2] to define and describe hate speech, offensive speech and free speech for annotation tasks. Hate speech (HT) in the South African context was defined as any unfairly discriminatory expression that demonstrates a clear intention to be harmful or to incite harm; promote or propagate hatred against a person or group of persons. Offensive speech (OFF) was defined as any fair or unfair expression that is not hate speech but discriminatory against a person or group of persons, while free speech (FS) was any expression that justifies the freedom of expressions' right. The FS was neither hate speech nor offensive speech.

About 21K tweets were initially collected but upon removal of duplicates, irrelevant and incorrectly annotated tweets, 14,896 resulted. During annotation, the tweets were divided according to the language of discourse into 7,100 for (English, English + Afrikaans), 4,102 for (English, English + Sesotho) and 4,500 for (English, English + IsiZulu) and two experienced human annotators, who were literate in each category of language labelled the tweets. By applying Cohen Kappa Agreement Test [8] for inter-annotator reliability, agreement scores of 0.837, 0.579 and 0.633, respectively were obtained. The resulting tweets were then pre-processed, as indicated in [7] since tweets can be very noisy and could distort the findings of the experiment.

The results showed that RF with word N-gram was the most effective in terms of accuracy of 0.893; SVM with character N-gram was the most effective in terms of true positive rate of 0.894 for HT detection; and GB with word N-gram was the most effective with true positive rate of 0.867 for OFF detection. By applying stacked ensemble generalization with multi-tier meta-features (multi-tier stacking ensemble), the true positive rates for HT and OFF were improved. However, the F1-scores remained poor with the best score of 0.56.

Therefore, this work explores ways of improving the performance, most especially F1-score by exploring neural embeddings and transformers.

3 Related Works

Literature has not reported the application of neural embeddings and transformer models for isolation of South African offensive and hate speech, but they have been employed in other contexts.

3.1 Neural Embeddings

Wang *et al.* [9] evaluated different embeddings models including skip-gram and continuous bag of words (CBOW) embeddings based on Word2Vec, GloVe, FastText, N-Gram2Vec and Dict2Vec for different applications using CNN and bidirectional LSTM. The intrinsic and extrinsic evaluators found skip-gram (Word2Vec) to be mostly efficient and outstanding in terms of accuracy for both neural networks.

Kumar *et al.* [10] employed GloVe pre-trained word embeddings, which included six billion tokens at 100 dimensions each to create a dense embedded matrix. They found that CNN + bidirectional LSTM ensemble classifier outperformed other deep learning neural networks such as CNN and bidirectional LSTM.

A classification experiment based on racist, sexist and neither classes was carried out using 16K annotated tweets in Badjatiya *et al.* [11]. The experiment made use of bag of word, TF-IDF weighted features, random embeddings as well as pre-trained GloVe and FastText character embeddings. From classification with several classifiers (LR, RF, SVM, GB, CNN and LSTM), the results showed that the random embedding with combination of LSTM and GB was the best with F-score of 0.93.

In Wang and Koopman [12], Word2Vec word embedding model outperformed pre-trained GloVe and Ariadne with an accuracy score of 0.627. An investigation into document embeddings also found Doc2Vec to be better than Ariadne, with least recall of 0.826 for both information classes.

3.2 Transformer-Based Deep Learning Models

In our model, BERT based neural transformer models are considered for their good performance [13]. Mozafari *et al.* [14] evaluated BERT on Waseem's racism and sexism dataset and Davidson's hate and offensive dataset. They employed CLS, non-linear layer, CNN and LSTM classifiers. While all the classifiers recorded good performances, CNN was the most outstanding with F1-score of 0.92. Sohn and Lee [15] developed a multichannel BERT for Classification of three popular hate speech datasets. The multichannel BERT consisted of mBERT, BERT and Chinese BERT, which have been pretrained and fine-tuned with pooling weight. The results of accuracy and F1-score were better than the individual models.

The reviewed works have shown that neural embeddings such as Word2Vec, Doc2Vec, GloVe and BERT transformer models performed well for different instances of hate and offensive speech isolation. In this work, however, both GloVe and BERT are not considered to avoid of the issue of out-of-vocabulary which might emanate due to the presence of many South African indigenous words in the evaluation dataset. This study, therefore, analyzes classical machine learning and neural network classifiers based on Word2Vec, Doc2Vec and mBERT.

4 Research Methods

4.1 Features Description

Firstly, two neural embeddings-based vector spaces were constructed from the vocabularies of the corpus using both Word2Vec and Doc2Vec models. Secondly, Multilingual Bidirectional Encoder Representations from Transformers (mBERT) was used to evaluate transformer-based model.

4.1.1 Word2Vec

Word2Vec [16] consists of a neural network architecture of an input layer, a projection layer and an output layer. It defines the probability of observing the target word w_t in a document D given its local context c_t.

Mikolov *et al.* [16] introduced the Skip-gram model which is an efficient method for learning vector representations of words from unstructured text data. Unlike previous neural network embeddings before it, the training does not involve dense matrix multiplications. The training objective is to find word representations that are useful for predicting the surrounding words in a sentence or a document.

Given a sequence of training words w_1, w_2, w_3,…, w_T, skip-gram model objective is to maximize the average log probability which is:

$$\frac{1}{T}\sum\nolimits_{t=1}^{T}\sum\nolimits_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{1}$$

where c is the size of the context.

The basic Skip-gram formulation defines $p(w_{t+j}\,|w_t)$ using the softmax function:

$$\mathrm{p}(w_o|w_I) = \frac{\exp(v_{WO}^{I}T_{V_{W1}})}{\sum_{w=1}^{W}\exp(v_{W}^{I}T_{V_{W1}})} \tag{2}$$

where v_W is the input and v'_W is the output vector representations of w, and W is the number of words in the vocabulary. Because of the impracticality of softmax function, $p(w_O|w_I)$ is replaced by negative sampling as:

$$log\sigma\left(v_{wo}^{I}T_{V_{W1}}\right)+\sum\nolimits_{i=1}^{k}\mathbb{E}_{wi\sim P_n(w)}[log\sigma\left(-v_{wi}^{I}T_{V_{W1}}\right)] \tag{3}$$

4.1.2 Doc2Vec

Doc2Vec [17], an extension of Word2Vec forms document representation by averaging word embeddings of all the words in a document. Doc2Vec consists of an input layer, a projection layer, and an output layer. The embeddings include target word and neighbouring words, which form the local contexts and the entire documents serving as the global contexts. Doc2Vec makes use of two approaches to learn paragraph and document representations, namely the distributed memory (DM) model and the distributed bag-of-words (DBOW) model.

Doc2Vec defines the probability of observing a target word w_t given its local context c_t as well as the global context X as

$$P\left(w^t|c^t,X\right)=\frac{\exp(v_{w^t}^{T}\left(U_{c^t}+\frac{1}{T}UX\right))}{\sum_{w^I\in V}\exp(U_{c^t}+\frac{1}{T}UX)} \tag{4}$$

U_{c^t} is the local context and $\frac{1}{T}UX$ is the global context and T is the length of the document. Thereafter, negative sampling is applied as presented in Word2Vec. The projection matrix from input space to hidden space U and projection matric from hidden space to output V are then learned to minimize the loss

$$l=-\sum\nolimits_{t=1}^{n}\sum\nolimits_{t=1}^{T_i}f(w_i^t,c_i^t,X_i^t) \tag{5}$$

Given the learned projection matrix U, each document is represented simply as an average of the embeddings of the words in the document.

$$d=\frac{1}{T}\sum\nolimits_{w\in D}U_w \tag{6}$$

The DBOW approach, which predicts target word based on contexts is used in this work.

4.1.3 Multilingual Bidirectional Encoder Representations from Transformers

Multilingual Bidirectional Encoder Representations from Transformers (mBERT) is the multilingual version of Bidirectional Encoder Representations from Transformers (BERT) [13], which is a bidirectional transformer encoder pretrained on English Wikipedia and Book Corpus. Unlike word embeddings, BERT models capture the contexts of texts, an improvement over existing contextual embeddings that learn either left-wise or right-wise. It makes use of masked language model to beat the challenge of bidirectional learning before and the next sentence prediction step. The pre-trained

BERT is fine-tuned by passing the word inputs through 12-layers of transformer encoder consisting of self-attention, add/normalise and feed-forward processes. The fine-tuned embeddings go through the neural prediction layer to produce the outputs. Figure 1 presents the architecture of the BERT.

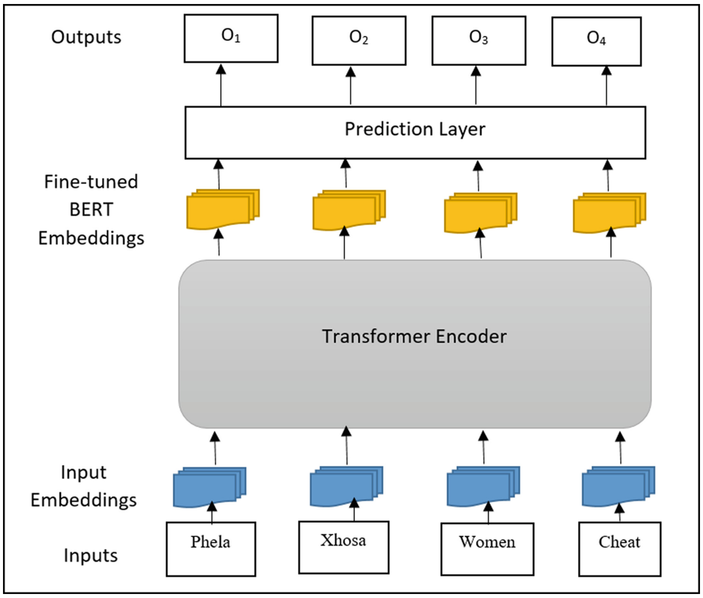

Fig. 1. The BERT architecture

The mBERT is used in this research because of its ability to accommodate different languages including Afrikaans, Sesotho and IsiZulu.

4.2 Classification Models

The classical machine learning algorithms such as *Logistic Regression* (LogReg), *Suport Vector Machine* (SVM), *Random Forest* (RF) and *Gradient Boosting* (GB) were evaluated and compared with the deep learning algorithms such as *Convolutional Neural Network* (CNN), *Long Short-term Memory* (LSTM), Bidirectional *Long Short-term Memory* (LSTM), *Gated Recurrent Unit* (GRU), Bidirectional *Gated Recurrent Unit* (GRU) and their various ensembles. We made use of Python's scikit-learn library by Pedregosa *et al.* [18] for the classical machine learning modelling and Keras by Chollet [19] for the deep learning modelling.

5 Experimental Setup

5.1 Dataset Description

Since the paper builds on the previous works, the models were evaluated using the abusive language dataset[1] used in [7]. The total and average distribution of words and characters in the dataset is presented in Table 1. The distribution shows that there were 28,912 unique words from which English words was about 28 times the size of South African words.

Table 1. Corpus statistics

Parameters	Values
Total number of unique words	28,912
Number of unique English words	27,466
Number of unique non-English South African words	1,446
Average number of words per tweet	23.76
Average length of words per tweet	142.75
Total number of tweets	14,896

5.2 Pre-processing

The tweets were cleaned and formatted by first tokenizing the tweets using Tweet-Tokenizer of Natural Language Processing Toolkit (NLTK) [20] before stemming the English words using WordNet Lemmatizer in NLTK [20]. Unwanted text such as username, punctuations, emoticons, emojis, special characters and hash symbols in hash tags were also removed. To avoid the omission of important terms as stop words, only English stopwords which are not present in other South African languages are extracted. We finally changed all text to lower case format.

5.3 Implementation

After implementing the pre-processing procedure using NLTK 3.5 [21], the dataset ($n = 14{,}896$) was divided into training ($n = 11{,}172$) and testing samples ($n = 3{,}724$) for the classification tasks. The parameter settings and their values for classical machine learning models are presented in Table 2. The selection of the best machine learning models was done using an exhaustive grid-search optimization over several hyperparameter configurations as highlighted in [7], while the Deep Leaning models were developed using 100 epochs, batch size of 64 and early stopping criteria (validation = loss, mode = min, verbose = 1, patience = 10, min_delta = 0.0001) without exhaustive grid-search due to extensive computing power requirements.

[1] https://github.com/oriolao/AbusiveLanguage.

Gensim [21] was used to implement the Word2Vec and Doc2Vec. For the Word2Vec, we used the skip-gram model [22] to create a 300-dimensional dense vector space. The model was implemented with minimum word count of 1, context window size of 10, workers of 8 and epoch of 30. For the Doc2Vec, we used DBOW (sg = 0) [12], which is built on a skip-gram model to create a 300-dimensional dense vector space. The parameters of Doc2Vec included negative, which set at 20 and learning rate, which is set at 0.002. The two embeddings were oversampled using synthetic minority oversampling technique (SMOTE) [23] to avoid imbalance in training.

In addition, *BERT-Base, Multilingual Cased Type* of transformer model with 104 languages, 12-layer, 768-hidden, 12-heads and 178,446,340 (170M) parameters was used to model mBERT. It was evaluated with batch size of 16bits, epoch of 5 and learning rate of 0.00001. The Python codes can be accessed here[2].

5.4 Performance Evaluation

The performance of the models was evaluated using accuracy and macro-averaging F1-score ($F1_m$). Also, macro-averaging precision ($precision_m$) and macro-averaging recall ($recall_m$) scores were evaluated for thorough comparison with the previous work [7]. Only the macro-averaging metrics were employed in the evaluation because the scores of the micro-averaging scores are always identical to the accuracy in multiclass classification. The equations for estimating accuracy, $\mathrm{F1_m}$, $\mathrm{precision_m}$ and $\mathrm{recall_m}$ are presented in Eq. 7, 8, 9, 10.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{7}$$

$$\mathrm{F1_m} = \frac{2\mathrm{X}(Recall_m \mathrm{X} Precision_m)}{(Recall_m + Precision_m)} \tag{8}$$

$$\mathrm{Precision_m} = \frac{\sum_{i=1}^{l} \frac{\text{TP}}{\text{TP+FP}}}{l} \tag{9}$$

$$\mathrm{Recall_m} = \frac{\sum_{i=1}^{l} \frac{\text{TP}}{\text{TP+FN}}}{l} \tag{10}$$

where TP is true positive; TN is true negative; FP is false positive; FN is false negative; and l is the number of classes.

6 Results

The test results for the $F1_m$ and accuracy performances of Word2Vec, Doc2Vec and mBERT based on classical and neural networks are presented in Table 2.

[2] https://github.com/oriolao/AbusiveLanguage.

Table 2. Test results for $F1_m$ and accuracy of the models

Type of model	Semantic embedding	Machine learning model	$F1_m$	Accuracy
Classical Machine Learning Model	Word2Vec	LogReg	0.31	0.884
		SVM	0.62	0.882
		RF	0.59	0.885
		GB	0.58	0.819
	Doc2Vec	LogReg	0.31	0.884
		SVM	0.44	0.597
		RF	0.58	0.884
		GB	0.58	0.744
Neural Networks	Word2Vec	MLP	0.44	0.577
		CNN	0.31	0.501
		LSTM	0.35	0.815
		GRU	0.37	0.653
		BiLSTM	0.33	0.647
		BiGRU	0.41	0.752
		Double CNN	0.38	0.872
		Double LSTM	0.43	0.748
		Double GRU	0.36	0.840
		CNN + LSTM	0.36	0.784
		CNN + GRU	0.41	0.850
	Doc2Vec	MLP	0.31	0.884
		CNN	0.36	0.800
		LSTM	0.32	0.695
		GRU	0.25	0.432
		BiLSTM	0.28	0.482
		BiGRU	0.09	0.145
		Double CNN	0.24	0.350
		Double LSTM	0.06	0.093
		Double GRU	0.24	0.403
		CNN + LSTM	0.27	0.440
		CNN + GRU	0.29	0.468
	Transformer model	BERT	0.46	0.833

The results show that Word2Vec with SVM recorded the highest F1$_m$ of 0.62, followed by Word2Vec with RF of 0.59. Both Doc2Vec with RF and GB recorded F1$_m$ of 0.58 each. Word2Vec with GB also recorded accuracy of 0.58. The performances of Word2Vec and Doc2Vec for all the neural models were poor with the highest of 0.44 recorded by MLP. The neural network models including deep learning models performed less impressively in terms of F1$_m$

In terms of accuracy, Word2Vec and Doc2Vec, with RF were impressive with accuracy scores of 0.885 and 0.884, respectively. Also, Word2Vec with SVM performed well with accuracy of 0.882. Doc2Vec with GB recorded an accuracy of 0.744. Both SVM and RF also performed effectively with little bias for the majority class. The mBERT accuracy was however moderate at 0.833.

In comparing the performance of the current models with the previous models [7], accuracy and TPR performances are presented in Table 3 and F1m, precision$_{\mathrm{m}}$ and recall$_{\mathrm{m}}$ are presented in Table 4. From Table 3, the current models performed worse than the previous models in terms of accuracy but the TPR for the current machine learning models, most especially SVM were closer to the previous models. The best of each type of models are indicated in boldface.

Table 3. Comparison of Accuracy and TPR of the current and previous models

Model	Feature	Type	Machine Leaning Model	TPR			Accuracy
				HT	OFF	FS	All
Previous Model[7]	N-gram surface level features	Classical	SVM	**0.894**	0.069	0.864	0.646
			RF	0.094	0.631	**0.966**	**0.893**
			GB	0.529	0.867	0.804	0.803
		Baseline solution	Multitier stacking ensemble	0.858	**0.887**	0.646	0.674
Current Model	Neural Networks	Word2Vec	SVM	0.376	0.738	0.910	0.882
			RF	0.224	0.646	**0.928**	**0.885**
			GB	0.576	0.804	0.827	0.819
		Doc2Vec	SVM	**0.718**	**0.827**	0.570	0.597
			RF	0.235	0.637	0.927	0.884
			GB	0.682	0.778	0.742	0.744
		Transformer model	BERT	0.000	0.31 2	0.687	0.833

The results in Table 4 show that the current models with the highest F1$_m$ of 0.62 outperformed the previous models that recorded the highest F1$_m$ of 0.56. The precision$_m$ and recall$_m$ of the current models were worse than the previous models. The results in

Table 3 and Table 4 also show that Word2Vec was slightly better than Doc2Vec. The best of each type of models are indicated in boldface.

The Cochrans-q test [24] was conducted on both the predictions of Word2Vec and the baseline solution. The results showed that at 0.05 significance, p-value of 0 (Q = 1854.77) was obtained meaning there was no significance difference. This shows that the current model is comparable to the previous model in terms of accuracy.

Table 4. Comparison of $\text{Precision}_\text{m}$, Recall_m and F1_m for the proposed statistical models and previous models

Model	Feature	Type	Machine Leaning Model	Precision_m	Recall_m	F1_m
Previous Model [7]	N-gram surface level features	Classical	SVM	**0.65**	0.55	0.35
			RF	0.55	0.47	0.50
			GB	0.52	0.73	**0.56**
		Baseline	Multitier Stacking Ensemble	0.50	**0.80**	0.50
Current Model	Neural Networks	Word2Vec	SVM	**0.58**	0.67	**0.62**
			RF	0.57	0.61	0.59
			GB	0.53	**0.74**	0.58
		Doc2Vec	SVM	0.46	0.70	0.44
			RF	0.57	0.60	0.58
			GB	0.50	0.73	0.58
		Transformer model	BERT	0.44	0.48	0.46

7 Conclusion

In this work, we have explored neural embedding and transformer models and compared their performances to our previous work [7]. The aim of the current work was to improve the F1-score in the classification of abusive languages for the purpose of effective isolation of offensive and hate speech. Based on the review of neural-based embedding and transformer models, we employed Word2Vec, Doc2vec and multilingual Bidirectional Encoder Representations from Transformers to classify abusive South African tweets into hate, offensive and free speech. Using the existing South African election dataset of 14K, the classical machine learning models such as Logistic Regression, Support Vector Machine, Random Forest and Gradient Boosting were first used to evaluate the neural features. Thereafter, different models of neural networks were explored.

With the exception of Logistic Regression, the results showed that the classical machine learning models based on Word2Vec was impressive with F1-score performance of 0.62 as against 0.56 of the previous model [7]. In terms of accuracy, a baseline performance of 0.885, which is less than the previous model by negligible value of 0.008 was recorded. On the other hand, the deep learning models performed worse than the classical machine learning models but the multilayer perceptron neural networks recorded high true positive rate for hate speech and offensive speech. The outcomes of the experiment showed that Word2Vec with Support Vector Machine performed best in terms of F1-score. In future work, Word2Vec will be enhanced through feature augmentation.

References

1. Georgakopoulos, S.V., Tasoulis, S.K., Vrahatis, A.G., Plagianakos, V.P.: Convolutional neural networks for toxic comment classification. In: the 10th Hellenic Conference (2018). https://doi.org/10.48550/arxiv.1802.09957
2. Prevention and Combating of Hate Crimes and Hate Speech Bill. (2018). https://www.justice.gov.za/legislation/hcbill/B9-2018-HateCrimesBill.pdf
3. PeaceTech Lab: Monitoring and Analysis of Hateful Language in South Africa Report #6. *PeaceTech Lab* (2019). https://www.peacetechlab.org/south-africa-report-6. Accessed: 15 Nov 2019
4. Mehdad, Y., Tetreault, J.: Do characters abuse more than words? In: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue 299–303 (Association for Computational Linguistics) (2016). https://doi.org/10.18653/v1/w16-3638
5. Fortuna, P., Nunes, S.: A survey on automatic detection of hate speech in text. ACM Comput. Surv. **51**(4), 51 (2018)
6. Parihar, A.S., Thapa, S., Mishra, S.: Hate speech detection using natural language processing : applications and hate speech detection using natural language processing: applications and challenges. In: Proceedings of the Fifth International Conference on Trends in Electronics and Informatics (ICOEI) 1302–1308. IEEE (2021). https://doi.org/10.1109/ICOEI51242.2021.9452882
7. Oriola, O., Kotzé, E.: Evaluating machine learning techniques for detecting offensive and hate speech in South African tweets. IEEE Access **8**, 21496–21509 (2020)
8. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Meas. **20**, 37–46 (1960)
9. Wang, B., Wang, A., Chen, F., Wang, Y., Kuo, C.: Evaluating word embedding models: methods and experimental results. APSIPA Trans. Signal and Inf. Process. **8**(E19), 1–13 (2019). https://doi.org/10.1017/ATSIP.2019.12
10. Kumar, S.: Fake news detection using deep learning models : a novel approach. Trans Emerg. Tel Tech. **31**(2), 1–23 (2019). https://doi.org/10.1002/ett.3767
11. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep Learning for Hate Speech Detection in Tweets. In: International World Wide Web Conference Committee (IW3C2), pp. 759–760 (2017)
12. Wang, S., Koopman, R.: Semantic embedding for information retrieval. In: BIR 2017 Workshop on Bibliometric-enhanced Information Retrieval, pp. 122–132 (2017)
13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186 (2019)

14. Mozafari, M., Reza Farahbakhsh, R., Crespi, N.: A BERT-based transfer learning approach for hate speech detection in online social media. In: 8th International Conference on Complex Networks and their Applications (2019). https://doi.org/10.48550/arXiv.1910.12574
15. Sohn, H., Lee, H.: MC-BERT4HATE: hate speech detection using multi-channel bert for different languages and translations. In: 2019 International Conference on Data Mining Workshops (ICDMW) 551–559. IEEE (2019). https://doi.org/10.1109/ICDMW.2019.00084
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
17. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31 st International Conference on Machine Learning 32, pp. 1188–1196 (2014). https://proceedings.mlr.press/v32/le14.html
18. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
19. Chollet, F.: *Deep Learning with Python.* (Simon and Schuster, 2021)
20. Bird, S., Kliein, E., Loper, E.: Analyzing Texts with Natural Language Toolkit: Natural Language Processing with Python. (O'Reilly, 2009)
21. Radim, R., Valletta, P.S.: Software framework for topic modelling with large corpora. In: Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks, Pp. 46–50 (2010)
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, pp. 1–9 (2013). https://doi.org/10.48550/arXiv.1310.4546
23. Chawla, N.V, Bowyer, K.W., Hall, L.O.: SMOTE: synthetic minority over-sampling technique. J. Artificial Intelligence Res. **16**, 321–357 (2002)
24. Raschka, S.: MLxtend: providing machine learning and data science utilities and extensions to python' s scientific computing stack. J. Open Source Softw. **3**, 24–25 (2018)