A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media

Marzieh Mozafari[(✉)], Reza Farahbakhsh, and Noël Crespi

CNRS UMR5157, Télécom SudParis, Institut Polytechnique de Paris, Évry, France
{marzieh.mozafari,reza.farahbakhsh,noel.crespi}@telecom-sudparis.eu

Abstract. Generated hateful and toxic content by a portion of users in social media is a rising phenomenon that motivated researchers to dedicate substantial efforts to the challenging direction of hateful content identification. We not only need an efficient automatic hate speech detection model based on advanced machine learning and natural language processing, but also a sufficiently large amount of annotated data to train a model. The lack of a sufficient amount of labelled hate speech data, along with the existing biases, has been the main issue in this domain of research. To address these needs, in this study we introduce a novel transfer learning approach based on an existing pre-trained language model called BERT (Bidirectional Encoder Representations from Transformers). More specifically, we investigate the ability of BERT at capturing hateful context within social media content by using new fine-tuning methods based on transfer learning. To evaluate our proposed approach, we use two publicly available datasets that have been annotated for racism, sexism, hate, or offensive content on Twitter. The results show that our solution obtains considerable performance on these datasets in terms of precision and recall in comparison to existing approaches. Consequently, our model can capture some biases in data annotation and collection process and can potentially lead us to a more accurate model.

Keywords: Hate speech detection · Transfer learning · Language modeling · BERT · Fine-tuning · NLP · Social media

1 Introduction

People are increasingly using social networking platforms such as Twitter, Facebook, YouTube, etc. to communicate their opinions and share information. Although the interactions among users on these platforms can lead to constructive conversations, they have been increasingly exploited for the propagation of abusive language and the organization of hate-based activities [1,16], especially due to the mobility and anonymous environment of these online platforms. Violence attributed to online hate speech has increased worldwide. For example, in the UK, there has been a significant increase in hate speech towards the

immigrant and Muslim communities following the UK's leaving the EU and the Manchester and London attacks[1]. The US also has been a marked increase in hate speech and related crime following the Trump election[2]. Therefore, governments and social network platforms confronting the trend must have tools to detect aggressive behavior in general, and hate speech in particular, as these forms of online aggression not only poison the social climate of the online communities that experience it, but can also provoke physical violence and serious harm [16].

Recently, the problem of online abusive detection has attracted scientific attention. Proof of this is the creation of the third Workshop on Abusive Language Online[3] or Kaggle's Toxic Comment Classification Challenge that gathered 4,551 teams[4] in 2018 to detect different types of toxicities (threats, obscenity, etc.). In the scope of this work, we mainly focus on the term hate speech as abusive content in social media, since it can be considered a broad umbrella term for numerous kinds of insulting user-generated content. Hate speech is commonly defined as any communication criticizing a person or a group based on some characteristics such as gender, sexual orientation, nationality, religion, race, etc. Hate speech detection is not a stable or simple target because misclassification of regular conversation as hate speech can severely affect users' freedom of expression and reputation, while misclassification of hateful conversations as unproblematic would maintain the status of online communities as unsafe environments [2].

To detect online hate speech, a large number of scientific studies have been dedicated by using Natural Language Processing (NLP) in combination with Machine Learning (ML) and Deep Learning (DL) methods [1,8,11,13,22,25]. Although supervised machine learning-based approaches have used different text mining-based features such as surface features, sentiment analysis, lexical resources, linguistic features, knowledge-based features or user-based and platform-based metadata [3,6,23], they necessitate a well-defined feature extraction approach. The trend now seems to be changing direction, with deep learning models being used for both feature extraction and the training of classifiers. These newer models are applying deep learning approaches such as Convolutional Neural Networks (CNNs), Long Short-Term Memory Networks (LSTMs), etc. [1,8] to enhance the performance of hate speech detection models, however, they still suffer from lack of labelled data or inability to improve generalization property.

Here, we propose a transfer learning approach for hate speech understanding using a combination of the unsupervised pre-trained model BERT [4] and some new supervised fine-tuning strategies. As far as we know, it is the first time that such exhaustive fine-tuning strategies are proposed along with a generative pre-trained language model to transfer learning to low-resource hate speech languages and improve performance of the task. In summary:

---

[1] Anti-muslim hate crime surges after Manchester and London Bridge attacks (2017): https://www.theguardian.com.

[2] A.: Hate on the rise after Trump's election: http://www.newyorker.com.

[3] https://sites.google.com/view/alw3/home.

[4] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/.

- We propose a transfer learning approach using the pre-trained language model BERT learned on English Wikipedia and BookCorpus to enhance hate speech detection on publicly available benchmark datasets. Toward that end, for the first time, we introduce new fine-tuning strategies to examine the effect of different embedding layers of BERT in hate speech detection.
- Our experiment results show that using the pre-trained BERT model and fine-tuning it on the downstream task by leveraging syntactical and contextual information of all BERT's transformers outperforms previous works in terms of precision, recall, and F1-score. Furthermore, examining the results shows the ability of our model to detect some biases in the process of collecting or annotating datasets. It can be a valuable clue in using pre-trained BERT model for debiasing hate speech datasets in future studies.

2   Previous Works

Here, the existing body of knowledge on online hate speech and offensive language and transfer learning is presented.

Online Hate Speech and Offensive Language: Researchers have been studying hate speech on social media platforms such as Twitter [3], Reddit [12,14], and YouTube [15] in the past few years. The features used in traditional machine learning approaches are the main aspects distinguishing different methods, and surface-level features such as bag of words, word-level and character-level n-grams, etc. have proven to be the most predictive features [11,13,22]. Apart from features, different algorithms such as Support Vector Machines [10], Naive Baye [16], and Logistic Regression [3,22], etc. have been applied for classification purposes. Waseem et al. [22] provided a test with a list of criteria based on the work in Gender Studies and Critical Race Theory (CRT) that can annotate a corpus of more than $16k$ tweets as racism, sexism, or neither. To classify tweets, they used a logistic regression model with different sets of features, such as word and character n-grams up to 4, gender, length, and location. They found that their best model produces character n-gram as the most indicative features, and using location or length is detrimental. Davidson et al. [3] collected a $24K$ corpus of tweets containing hate speech keywords and labelled the corpus as hate speech, offensive language, or neither by using crowd-sourcing and extracted different features such as n-grams, some tweet-level metadata such as the number of hashtags, mentions, retweets, and URLs, Part Of Speech (POS) tagging, etc. Their experiments on different multi-class classifiers showed that the Logistic Regression with L2 regularization performs the best at this task. Malmasi et al. [10] proposed an ensemble-based system that uses some linear SVM classifiers in parallel to distinguish hate speech from general profanity in social media.

As one of the first attempts in neural network models, Djuric et al. [5] proposed a two-step method including a continuous bag of words model to extract paragraph2vec embeddings and a binary classifier trained along with the embeddings to distinguish between hate speech and clean content. Badjatiya et al. [1] investigated three deep learning architectures, FastText, CNN, and LSTM,

in which they initialized the word embeddings with either random or GloVe embeddings. Gambäck et al. [8] proposed a hate speech classifier based on CNN model trained on different feature embeddings such as word embeddings and character n-grams. Zhang et al. [25] used a CNN+GRU (Gated Recurrent Unit network) neural network model initialized with pre-trained word2vec embeddings to capture both word/character combinations (e. g., n-grams, phrases) and word/character dependencies (order information). Waseem et al. [23] brought a new insight to hate speech and abusive language detection tasks by proposing a multi-task learning framework to deal with datasets across different annotation schemes, labels, or geographic and cultural influences from data sampling. Founta et al. [7] built a unified classification model that can efficiently handle different types of abusive language such as cyberbullying, hate, sarcasm, etc. using raw text and domain-specific metadata from Twitter. Furthermore, researchers have recently focused on the bias derived from the hate speech training datasets [2,21,24]. Davidson et al. [2] showed that there were systematic and substantial racial biases in five benchmark Twitter datasets annotated for offensive language detection. Wiegand et al. [24] also found that classifiers trained on datasets containing more implicit abuse (tweets with some abusive words) are more affected by biases rather than once trained on datasets with a high proportion of explicit abuse samples (tweets containing sarcasm, jokes, etc.).

Transfer Learning: Pre-trained vector representations of words, embeddings, extracted from vast amounts of text data have been encountered in almost every language-based tasks with promising results. Two of the most frequently used context-independent neural embeddings are word2vec and Glove extracted from shallow neural networks. The year 2018 has been an inflection point for different NLP tasks thanks to remarkable breakthroughs: Universal Language Model Fine-Tuning (ULMFiT) [9], Embedding from Language Models (ELMO) [17], OpenAI's Generative Pre-trained Transformer (GPT) [18], and Google's BERT model [4]. Howard et al. [9] proposed ULMFiT which can be applied to any NLP task by pre-training a universal language model on a general-domain corpus and then fine-tuning the model on target task data using discriminative fine-tuning. Peters et al. [17] used a bi-directional LSTM trained on a specific task to present context-sensitive representations of words in word embeddings by looking at the entire sentence. Radford et al. [18] and Devlin et al. [4] generated two transformer-based language models, OpenAI GPT and BERT respectively. OpenAI GPT [18] is an unidirectional language model while BERT [4] is the first deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus. BERT has two novel prediction tasks: Masked LM and Next Sentence Prediction. The pre-trained BERT model significantly outperformed ELMo and OpenAI GPT in a series of downstream tasks in NLP [4]. Identifying hate speech and offensive language is a complicated task due to the lack of undisputed labelled data [10] and the inability of surface features to capture the subtle semantics in text. To address this issue, we use the pre-trained language model BERT for hate speech classification and try to fine-tune specific task by leveraging information from different transformer encoders.

3   Methodology

Here, we analyze the BERT transformer model on the hate speech detection task. BERT is a multi-layer bidirectional transformer encoder trained on the English Wikipedia and the Book Corpus containing 2,500M and 800M tokens, respectively, and has two models named $BERT_{base}$ and $BERT_{large}$. $BERT_{base}$ contains an encoder with 12 layers (transformer blocks), 12 self-attention heads, and 110 million parameters whereas $BERT_{large}$ has 24 layers, 16 attention heads, and 340 million parameters. Extracted embeddings from $BERT_{base}$ have 768 hidden dimensions [4]. As the BERT model is pre-trained on general corpora, and for our hate speech detection task we are dealing with social media content, therefore as a crucial step, we have to analyze the contextual information extracted from BERT' s pre-trained layers and then fine-tune it using annotated datasets. By fine-tuning we update weights using a labelled dataset that is new to an already trained model. As an input and output, BERT takes a sequence of tokens in maximum length 512 and produces a representation of the sequence in a 768-dimensional vector. BERT inserts at most two segments to each input sequence, [CLS] and [SEP]. [CLS] embedding is the first token of the input sequence and contains the special classification embedding which we take the first token [CLS] in the final hidden layer as the representation of the whole sequence in hate speech classification task. The [SEP] separates segments and we will not use it in our classification task. To perform the hate speech detection task, we use $BERT_{base}$ model to classify each tweet as Racism, Sexism, Neither or Hate, Offensive, Neither in our datasets. In order to do that, we focus on fine-tuning the pre-trained $BERT_{base}$ parameters. By fine-tuning, we mean training a classifier with different layers of 768 dimensions on top of the pre-trained $BERT_{base}$ transformer to minimize task-specific parameters.

3.1   Fine-Tuning Strategies

Different layers of a neural network can capture different levels of syntactic and semantic information. The lower layer of the BERT model may contain more general information whereas the higher layers contain task-specific information [4], and we can fine-tune them with different learning rates. Here, four different fine-tuning approaches are implemented that exploit pre-trained $BERT_{base}$ transformer encoders for our classification task. More information about these transformer encoders' architectures are presented in [4]. In the fine-tuning phase, the model is initialized with the pre-trained parameters and then are fine-tuned using the labelled datasets. Different fine-tuning approaches on the hate speech detection task are depicted in Fig. 1, in which X_i is the vector representation of token i in a tweet sample, and are explained in more detail as follows:

1. BERT based fine-tuning: In the first approach, which is shown in Fig. 1a, very few changes are applied to the $BERT_{base}$. In this architecture, only the [CLS] token output provided by BERT is used. The [CLS] output, which is equivalent to the [CLS] token output of the 12th transformer encoder, a vector of size 768, is given as input to a fully connected network without hidden layer. The softmax activation function is applied to the hidden layer to classify.
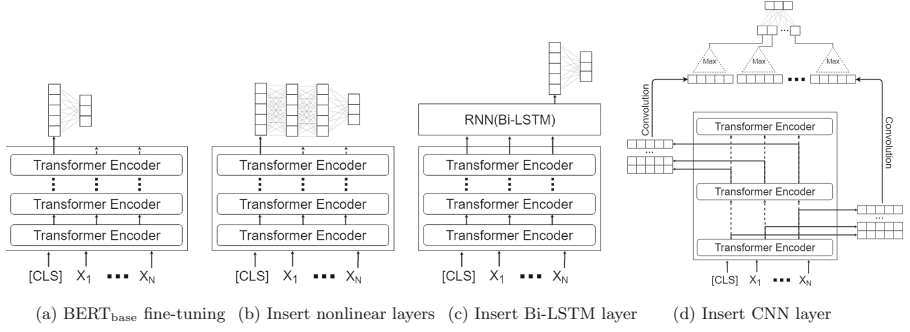
(a) BERT_base fine-tuning (b) Insert nonlinear layers (c) Insert Bi-LSTM layer (d) Insert CNN layer

Fig. 1. Fine-tuning strategies

2. Insert nonlinear layers: Here, the first architecture is upgraded and an architecture with a more robust classifier is provided in which instead of using a fully connected network without hidden layer, a fully connected network with two hidden layers in size 768 is used. The first two layers use the Leaky Relu activation function with negative slope = 0.01, but the final layer, as the first architecture, uses softmax activation function as shown in Fig. 1b.

3. Insert Bi-LSTM layer: Unlike previous architectures that only use [CLS] as the input for the classifier, in this architecture all outputs of the latest transformer encoder are used in such a way that they are given as inputs to a bidirectional recurrent neural network (Bi-LSTM) as shown in Fig. 1c. After processing the input, the network sends the final hidden state to a fully connected network that performs classification using the softmax activation function.

4. Insert CNN layer: In this architecture shown in Fig. 1d, the outputs of all transformer encoders are used instead of using the output of the latest transformer encoder. So that the output vectors of each transformer encoder are concatenated, and a matrix is produced. The convolutional operation is performed with a window of size (3, hidden size of BERT which is 768 in BERT_base model) and the maximum value is generated for each transformer encoder by applying max pooling on the convolution output. By concatenating these values, a vector is generated which is given as input to a fully connected network. By applying softmax on the input, the classification operation is performed.

4 Experiments and Results

We first introduce datasets used in our study and then investigate the different fine-tuning strategies for hate speech detection task. We also include the details of our implementation and error analysis in the respective subsections.

4.1 Dataset Description

We evaluate our method on two widely-studied datasets provided by Waseem and Hovey [22] and Davidson et al. [3]. Waseem and Hovy [22] collected 16k

of tweets based on an initial ad-hoc approach that searched common slurs and terms related to religious, sexual, gender, and ethnic minorities. They annotated their dataset manually as racism, sexism, or neither. To extend this dataset, Waseem [20] also provided another dataset containing $6.9k$ of tweets annotated with both expert and crowdsourcing users as racism, sexism, neither, or both. Since both datasets are overlapped partially and they used the same strategy in definition of hateful content, we merged these two datasets following Waseem et al. [23] to make our imbalance data a bit larger. Davidson et al. [3] used the Twitter API to accumulate 84.4 million tweets from 33,458 twitter users containing particular terms from a pre-defined lexicon of hate speech words and phrases, called Hatebased.org. To annotate collected tweets as Hate, Offensive, or Neither, they randomly sampled $25k$ tweets and asked users of CrowdFlower crowdsourcing platform to label them. In detail, the distribution of different classes in both datasets will be provided in Subsect. 4.3.

4.2   Pre-processing

We find mentions of users, numbers, hashtags, URLs and common emoticons and replace them with the tokens <user>, <number>, <hashtag>, <url>, <emoticon>. We also find elongated words and convert them into short and standard format; for example, converting yeeeessss to yes. With hashtags that include some tokens without any with space between them, we replace them by their textual counterparts; for example, we convert hashtag "#notsexist" to "not sexist". All punctuation marks, unknown uni-codes and extra delimiting characters are removed, but we keep all stop words because our model trains the sequence of words in a text directly. We also convert all tweets to lower case.

4.3   Implementation and Results Analysis

For the implementation of our neural network, we used pytorch-pretrained-bert library containing the pre-trained BERT model, text tokenizer, and pre-trained WordPiece. As the implementation environment, we use Google Colaboratory tool which is a free research tool with a Tesla K80 GPU and 12G RAM. Based on our experiments, we trained our classifier with a batch size of 32 for 3 epochs. The dropout probability is set to 0.1 for all layers. Adam optimizer is used with a learning rate of $2e-5$. As an input, we tokenized each tweet with the BERT tokenizer. It contains invalid characters removal, punctuation splitting, and lowercasing the words. Based on the original BERT [4], we split words to subword units using WordPiece tokenization. As tweets are short texts, we set the maximum sequence length to 64 and in any shorter or longer length case it will be padded with zero values or truncated to the maximum length.

We consider 80% of each dataset as training data to update the weights in the fine-tuning phase, 10% as validation data to measure the out-of-sample performance of the model during training, and 10% as test data to measure the out-of-sample performance after training. To prevent overfitting, we use stratified sampling to select 0.8, 0.1, and 0.1 portions of tweets from each class

Table 1. Dataset statistics of the both Waseem-dataset (a) and Davidson-dataset (b). Splits are produced using stratified sampling to select 0.8, 0.1, and 0.1 portions of tweets from each class (racism/sexism/neither or hate/offensive/neither) for train, validation, and test samples, respectively.

	Racism	Sexism	Neither	Total
Train	1693	3337	10787	15817
Validation	210	415	1315	1940
Test	210	415	1315	1940
Total	2113	4167	13417	

	Hate	Offensive	Neither	Total
Train	1146	15354	3333	19832
Validation	142	1918	415	2475
Test	142	1918	415	2475
Total	1430	19190	4163	

(a) Waseem-dataset.

(b) Davidson-dataset.

(racism/sexism/neither or hate/offensive/neither) for train, validation, and test. Classes' distribution of train, validation, and test datasets are shown in Table 1.

As it is understandable from Table 1(a) and (b), we are dealing with imbalance datasets with various classes' distribution. Since hate speech and offensive languages are real phenomena, we did not perform oversampling or undersampling techniques to adjust the classes' distribution and tried to supply the datasets as realistic as possible. We evaluate the effect of different fine-tuning strategies on the performance of our model. Table 2 summarized the obtained results for fine-tuning strategies along with the official baselines. We use Waseem and Hovy [22], Davidson et al. [3], and Waseem et al. [23] as baselines and compare the results with our different fine-tuning strategies using pre-trained BERT$_{base}$ model. The evaluation results are reported on the test dataset and on three different metrics: precision, recall, and weighted-average F1-score. We consider weighted-average F1-score as the most robust metric versus class imbalance, which gives insight into the performance of our proposed models. According to Table 2, F1-scores of all BERT based fine-tuning strategies except BERT +

Table 2. Results on the trial data using pre-trained BERT model with different fine-tuning strategies and comparison with results in the literature.

Method	Datasets	Precision(%)	Recall(%)	F1-score(%)
Waseem and Hovy [22]	Waseem	72.87	77.75	73.89
Davidson et al. [3]	Davidson	91	90	90
Waseem et al. [23]	Waseem	-	-	80
	Davidson	-	-	89
BERT$_{base}$	Waseem	81	81	81
	Davidson	91	91	91
BERT$_{base}$ + Nonlinear Layers	Waseem	73	85	76
	Davidson	76	78	77
BERT$_{base}$ + LSTM	Waseem	87	86	86
	Davidson	91	92	92
BERT$_{base}$ + CNN	Waseem	**89**	**87**	**88**
	Davidson	**92**	**92**	**92**

nonlinear classifier on top of BERT are higher than the baselines. Using the pre-trained BERT model as initial embeddings and fine-tuning the model with a fully connected linear classifier ($BERT_{base}$) outperforms previous baselines yielding F1-score of 81% and 91% for datasets of Waseem and Davidson respectively. Inserting a CNN to pre-trained BERT model for fine-tuning on downstream task provides the best results as F1- score of 88% and 92% for datasets of Waseem and Davidson and it clearly exceeds the baselines. Intuitively, this makes sense that combining all pre-trained BERT layers with a CNN yields better results in which our model uses all the information included in different layers of pre-trained BERT during the fine-tuning phase. This information contains both syntactical and contextual features coming from lower layers to higher layers of BERT.

4.4   Error Analysis

Although we have very interesting results in term of recall, the precision of the model shows the portion of false detection we have. To understand better this phenomenon, in this section we perform a deep analysis on the error of the model. We investigate the test datasets and their confusion matrices resulted from the $BERT_{base}$ + CNN model as the best fine-tuning approach; depicted in Figs. 2 and 3. According to Fig. 2 for Waseem-dataset, it is obvious that the model can separate sexism from racism content properly. Only two samples belonging to racism class are misclassified as sexism and none of the sexism samples are misclassified as racism. A large majority of the errors come from misclassifying hateful categories (racism and sexism) as hatless (neither) and vice versa. 0.9% and 18.5% of all racism samples are misclassified as sexism and neither respectively whereas it is 0% and 12.7% for sexism samples. Almost 12% of neither samples are misclassified as racism or sexism. As Figure 3 makes clear for Davidson-dataset, the majority of errors are related to hate class where the model misclassified hate content as offensive in 63% of the cases. However, 2.6% and 7.9% of offensive and neither samples are misclassified respectively.
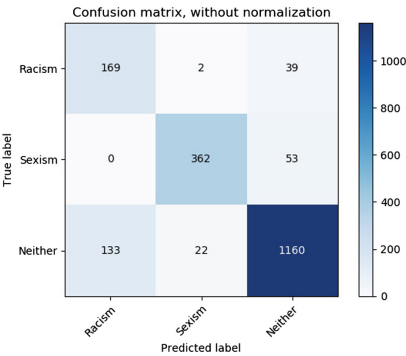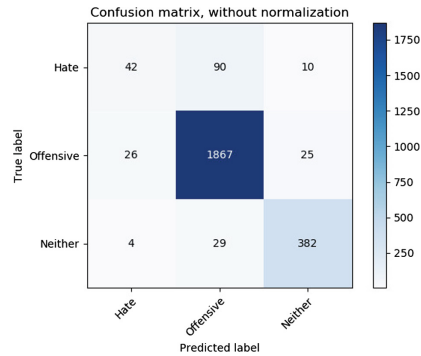

Fig. 2. Waseem-datase's confusion matrix

Fig. 3. Davidson-dataset's confusion matrix

Table 3. Misclassified samples from Waseem-dataset.

ID	Tweet	Annotated	Predicted
1	@user Good tweet. But they actually start selling their daughters at 9	Racism	Sexism
2	RT @user: Are we going to continue seeing the oppression of women or are we going to make a stand? #BanTheBurka http://t.co/hZDx8mlvTv	Racism	Sexism
3	RT @user: @user my comment was sexist, but I'm not personally, always a sexist	Sexism	Neither
4	RT @user: @user Ah, you're a #feminist? Seeing #sexism everywhere then, do check my tweets before you call me #sexist	Sexism	Neither
5	@user By hating the ideology that enables it, that is what I'm doing	Racism	Neither

Table 4. Misclassified samples from Davidson-dataset.

ID	Tweet	Annotated	Predicted
6	@user: If you claim Macklemore is your favorite rapper I'm also assuming you watch the WNBA on your free time faggot	Hate	Offensive
7	@user: Some black guy at my school asked if there were colored printers in the library. "It's 2014 man you can use any printer you want" I said	Hate	Neither
8	RT @user: @user typical coon activity	Hate	Neither
9	@user: @user @user White people need those weapons to defend themselves from the subhuman trash your sort unleashes on us	Neither	Hate
10	RT @user: Finally! Warner Bros. making superhero films starring a woman, person of color and actor who identifies as ""queer"";	Neither	Offensive

To understand better the mislabeled items by our model, we did a manual inspection on a subset of the data and record some of them in Tables 3 and 4. Considering the words such as "daughters", "women", and "burka" in tweets with IDs 1 and 2 in Table 3, it can be understood that our BERT based classifier is confused with the contextual semantic between these words in the samples and misclassified them as sexism because they are mainly associated to femininity. In some cases containing implicit abuse (like subtle insults) such as tweets with IDs 5 and 7, our model cannot capture the hateful/offensive content and therefore misclassifies. It should be noticed that even for a human it is difficult to discriminate against this kind of implicit abuses.

By examining more samples and with respect to recently studies [2,19,24], it is clear that many errors are due to biases from data collection [24] and rules of annotation [19] and not the classifier itself. Since Waseem et al. [22] created a small ad-hoc set of keywords and Davidson et al. [3] used a large crowdsourced dictionary of keywords (Hatebase lexicon) to sample tweets for training, they included some biases in the collected data. Especially for Davidson-dataset, some tweets with specific language (written within the African American Vernacular English) and geographic restriction (United States of America) are oversampled such as tweets containing disparage words "nigga", "faggot", "coon", or "queer", result in high rates of misclassification. However, these misclassifications do not confirm the low performance of our classifier because annotators tended to annotate many samples containing disrespectful words as hate or offensive without any presumption about the social context of tweeters such as the speaker's identity or dialect, whereas they were just offensive or even neither tweets. Tweets IDs 6, 8, and 10 are some samples containing offensive words and slurs which

arenot hate or offensive in all cases and writers of them used this type of language in their daily communications. Given these pieces of evidence, by considering the content of tweets, we can see in tweets IDs 3, 4, and 9 that our BERT-based classifier can discriminate tweets in which neither and implicit hatred content exist. One explanation of this observation may be the pre-trained general knowledge that exists in our model. Since the pre-trained BERT model is trained on general corpora, it has learned general knowledge from normal textual data without any purposely hateful or offensive language. Therefore, despite the bias in the data, our model can differentiate hate and offensive samples accurately by leveraging knowledge-aware language understanding that it has and it can be the main reason for high misclassifications of hate samples as offensive (in reality they are more similar to offensive rather than hate by considering social context, geolocation, and dialect of tweeters).

5   Conclusion

Conflating hatred content with offensive or harmless language causes online automatic hate speech detection tools to flag user-generated content incorrectly. Not addressing this problem may bring about severe negative consequences for both platforms and users such as decreasement of platforms' reputation or users abandonment. Here, we propose a transfer learning approach advantaging the pre-trained language model BERT to enhance the performance of a hate speech detection system and to generalize it to new datasets. To that end, we introduce new fine-tuning strategies to examine the effect of different layers of BERT in hate speech detection task. The evaluation results indicate that our model outperforms previous works by profiting the syntactical and contextual information embedded in different transformer encoder layers of the BERT model using a CNN-based fine-tuning strategy. Furthermore, examining the results shows the ability of our model to detect some biases in the process of collecting or annotating datasets. It can be a valuable clue in using the pre-trained BERT model to alleviate bias in hate speech datasets in future studies, by investigating a mixture of contextual information embedded in the BERT's layers and a set of features associated to the different type of biases in data.

References

1. Badjatiya, P., Gupta, S., Gupta, M., et al.: Deep learning for hate speech detection in tweets. CoRR **abs/1706.00188** (2017). http://arxiv.org/abs/1706.00188
2. Davidson, T., Bhattacharya, D., Weber, I.: Racial bias in hate speech and abusive language detection datasets. CoRR **abs/1905.12516** (2019). http://arxiv.org/abs/1905.12516
3. Davidson, T., Warmsley, D., Macy, M.W., et al.: Automated hate speech detection and the problem of offensive language. CoRR **abs/1703.04009** (2017). http://arxiv.org/abs/1703.04009
4. Devlin, J., Chang, M., Lee, K., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018). http://arxiv.org/abs/1810.04805

5. Djuric, N., Zhou, J., Morris, R., et al.: Hate speech detection with comment embeddings. In: Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Companion, pp. 29–30. ACM, New York (2015). https://doi.org/10.1145/2740908.2742760

6. Fortuna, P., Nunes, S.: A survey on automatic detection of hate speech in text. ACM Comput. Surv. **51**(4), 85:1–85:30 (2018). https://doi.org/10.1145/3232676

7. Founta, A.M., Chatzakou, D., Kourtellis, N., et al.: A unified deep learning architecture for abuse detection. In: Proceedings of the 10th ACM Conference on Web Science, WebSci 2019, pp. 105–114. ACM, New York (2019)

8. Gambäck, B., Sikdar, U.K.: Using convolutional neural networks to classify hate-speech. In: Proceedings of the First Workshop on Abusive Language Online, pp. 85–90. Association for Computational Linguistics, Vancouver (2017). https://doi.org/10.18653/v1/W17-3013

9. Howard, J., Ruder, S.: Fine-tuned language models for text classification. CoRR **abs/1801.06146** (2018). http://arxiv.org/abs/1801.06146

10. Malmasi, S., Zampieri, M.: Challenges in discriminating profanity from hate speech. CoRR **abs/1803.05495** (2018). http://arxiv.org/abs/1803.05495

11. 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, pp. 299–303. Association for Computational Linguistics, Los Angeles (2016). https://doi.org/10.18653/v1/W16-3638

12. Mittos, A., Zannettou, S., Blackburn, J., et al.: And We Will Fight For Our Race! A Measurement Study of Genetic Testing Conversations on Reddit and 4chan. CoRR **abs/1901.09735** (2019). http://arxiv.org/abs/1901.09735

13. Nobata, C., Tetreault, J., Thomas, A., et al.: Abusive language detection in online user content. In: Proceedings of the 25th International Conference on World Wide Web, WWW 2016, pp. 145–153. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2016). https://doi.org/10.1145/2872427.2883062

14. Olteanu, A., Castillo, C., Boy, J., et al.: The effect of extremist violence on hateful speech online. CoRR **abs/1804.05704** (2018). http://arxiv.org/abs/1804.05704

15. Ottoni, R., Cunha, E., Magno, G., et al.: Analyzing right-wing Youtube channels: hate, violence and discrimination. In: Proceedings of the 10th ACM Conference on Web Science, WebSci 2018, pp. 323–332. ACM, New York (2018). https://doi.org/10.1145/3201064.3201081

16. Pete, B., Williams, M.L.: Cyber hate speech on Twitter: an application of machine classification and statistical modeling for policy and decision making. Policy Internet **7**(2), 223–242 (2015). https://doi.org/10.1002/poi3.8

17. Peters, M.E., Neumann, M., Iyyer, M., et al.: Deep contextualized word representations. CoRR **abs/1802.05365** (2018). http://arxiv.org/abs/1802.05365

18. Radford, A.: Improving language understanding by generative pre-training (2018)

19. Sap, M., Card, D., Gabriel, S., et al.: The risk of racial bias in hate speech detection. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1668–1678. Association for Computational Linguistics, Florence (2019). https://doi.org/10.18653/v1/P19-1163

20. Waseem, Z.: 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, pp. 138–142. Association for Computational Linguistics, Austin (2016). https://doi.org/10.18653/v1/W16-5618

21. Waseem, Z., Davidson, T., Warmsley, D., et al.: Understanding abuse: a typology of abusive language detection subtasks. In: Proceedings of the First Workshop on Abusive Language Online, pp. 78–84. Association for Computational Linguistics, Vancouver (2017). https://doi.org/10.18653/v1/W17-3012, https://www.aclweb.org/anthology/W17-3012

22. Waseem, Z., Hovy, D.: Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In: Proceedings of the NAACL Student Research Workshop, pp. 88–93. Association for Computational Linguistics, San Diego (2016). https://doi.org/10.18653/v1/N16-2013

23. Waseem, Z., Thorne, J., Bingel, J.: Bridging the Gaps: Multi Task Learning for Domain Transfer of Hate Speech Detection, pp. 29–55. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-78583-7_3

24. Wiegand, M., Ruppenhofer, J., Kleinbauer, T.: Detection of abusive language: the problem of biased datasets. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 602–608. Association for Computational Linguistics, Minneapolis (2019). https://doi.org/10.18653/v1/N19-1060

25. Zhang, Z., Robinson, D., Tepper, J.: Detecting hate speech on twitter using a convolution-GRU based deep neural network. In: The Semantic Web, pp. 745–760. Springer International Publishing, Cham (2018)