Hate Speech Detection Based on Multiple Machine Learning Algorithms

Jialin Lu

Computer Science, University of British Columbia, Vancouver, V6T1Z4, Canada
jialinlu@student.ubc.ca

Abstract. Social media platforms such as Facebook, Twitter, and Reddit have experienced a substantial surge in user base and popularity over the past decade, facilitating global connectivity among billions of individuals. The major platforms have also served as a place for users to freely spread hate speech, which can be defined as offensive language against a specific group of people. Online hate speech has become a serious issue in the social media platforms, and can lead to negative psychological effects on the targeted people. Therefore, finding an effective model to classify a sequence as hate speech or not is very crucial. This paper treated this task as a sequence binary classification task, where the labels are hate speech and not hate speech, and conducted a comparative analysis on multiple different models with the binary label version of ETHOS dataset. Four metrics: accuracy, recall, precision, and F1 score were used to evaluate the trained/fine-tuned models, and the performance of each classification model that was trained/fine-tuned on ETHOS dataset were analyzed to discover potential weaknesses of the existing models. This research shows that the single-task fine-tuned BERT classifier resulted in the highest accuracy, recall, precision, and F1 score. Surprisingly, the simple probabilistic model Naïve Bayes also demonstrated good performance on hate speech classification using the test dataset. After thorough experimentation, this research also shows that the predictions of the Naïve Bayes and BiLSTM models are strongly affected by the appearance of words that are often associated and used in hate speech.

Keywords: Hate speech, Natural language processing, BERT.

1 Introduction

In the current era of heightened interconnectivity facilitated by prominent social media platforms such as Twitter and Instagram, a multitude of individuals numbering in the billions have gained unprecedented freedom and immediacy in expressing and disseminating their ideas and viewpoints [1]. However, social media platforms have often been misused as mediums to spread violent comments as well as hate speech [2]. Hate speech can be defined as the use of pejorative or offensive language that insults a person, or a group based on traits like race, ethnicity, religion, gender, sexual orientation, and nationality [3]. Studies have demonstrated that hate speech can cause

P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_25

multiple harmful phycological effects on the targeted groups, including but not limited to the LGBTQ community, thereby emphasizing the non-negligible adverse consequences associated with hate speech [2].

The mitigation of hate speech holds paramount significance for both social media corporations and their user base, given the detrimental consequences it engenders. However, the manual filtration of hate speech within the realm of messages, comments, or tweets is not only highly ineffectual but also lacks scalability owing to the colossal magnitude of social media users and the consequential demand for a substantial workforce to execute such a task. Therefore, it is crucial to automatically classify sentences from the large number of online contents as hate speech or non-hate speech. Machine learning techniques like Naïve Bayes and Random Forests have been demonstrated to perform classification tasks effectively due to their expeditious training process and capacity to generate accurate predictions that are readily comprehensible [4, 5]. However, deep learning models have shown higher performance in accuracy in various fields like computer vision and Natural Language Processing (NLP) in recent years. Recurrent Neural Network (RNN), which is designed to employ sequential data, has been demonstrated to be effective in various tasks e.g. named entity recognition and sentiment analysis [6, 7]. Moreover, the introduction of the Transformer architecture has revolutionized the field of NLP. With the encoder-decoder configuration and the attention mechanism, the Transformer model demonstrated excellent performance on translation task and showed its ability to generalize on other NLP tasks [8]. Furthermore, Devlin et al developed Bidirectional Encoder Representations from Transformers (BERT), which is a transformer-based model but built by stacking encoders only [9]. BERT has been pre-trained using the masked language model pre-training objective and next sentence prediction on large corpus taken from BookCorpus and Wikipedia. The pre-training allowed BERT to generate high quality representations for words which can be used in various downstream NLP tasks.

Previous studies have extensively investigated the detection of hate speech through the application of various deep learning methodologies and datasets in multiple languages, yielding promising outcomes. For instance, Rajput et al. compared the performance of deep learning models with different kinds of embeddings including GloVe, fastText, and static BERT embedding, and discovered that the static BERT embeddings outperformed the other kinds of embeddings [10]. Another study [11] on hate speech detection in Hindi and Marathi tweets has compared multiple deep learning architectures with FastText and random embeddings. The authors experimented with CNN, LSTM, and variations of BERT like Multilingual BERT and IndicBERT on the binary classification task. The paper also explored multi-lable dataset with four labels: hate speech, offensive language, profane words used, and none of the above. Furthermore, this study [12] compared the effectiveness of automatic feature selection and manual feature engineering without feature selection using a linear Support Vector Machine (SVM) as the classifier in hate speech detection. The feature selection process includes using Logistic Regression (LR) to calculate the importance score for each feature, and the classifier with feature selection resulted in a higher micro-F1 score on all datasets.

In order to conduct a more comprehensive analysis and comparison of the aforementioned models, this paper experiments with multiple machine learning architectures and compares their effectiveness on English hate speech detection. Hate speech detection is treated as a sequence binary classification task, where the labels are hate speech and no hate speech. The contribution of this paper includes analyzing the applications of different kinds of machine learning architectures including Naïve Bayes, Random Forest, BiLSTM, BERT on hate speech detection with data from YouTube and Reddit which are large platforms that hate speech is likely to spread in.

2 Method

2.1 Dataset

This paper employed a recent binary label data from ETHOS dataset which consists of 998 comments on Reddit and YouTube [13]. Among the comments, 433 of them are labeled as hate speech, and the other 565 comments are labeled as not hate speech. This dataset was chosen for this research is justified by its contemporaneity and relevance, as it provides an up-to-date representation of hate speech, capturing the evolving nature of online communication and the proliferation of new terminologies and internet slang. The dataset was split with 90% in training set and 10% in testing set.

2.2 Data Cleaning

Regular expression was firstly used to eliminate any non-English characters, numbers, and hashtags in the dataset. Stop words were not removed from the dataset since study has demonstrated that removing stop words in sequence classification task does not improve model performance and can even decrease the performance depending on the stop words removal method. The exclusion of stop words merely contributes a negligible reduction of approximately 1% in the feature space, namely the size of the vocabulary, while simultaneously potentially influencing the contextual interpretation of sentences in the context of classification tasks.

2.3 Model Architecture

This paper used 4 different classification models, namely Naïve Bayes, Random Forest, BiLSTM, and BERT to complete the task of hate speech detection which was treated as binary classification for sequence of words in the comments from the dataset. Adam optimizer and Categorical Cross Entropy loss were used for training BiLSTM and fine-tuning BERT. The initial learning rate was 0.001 for BiLSTM and 3×10^{-5} for BERT. The number of epochs the models were trained for that resulted in the maximum validation accuracy were chosen. A batch size of 64 was used for training BiLSTM and 32 for fine-tuning BERT. The detailed flow of the classification of these mentioned models can be found in Fig. 1 and Fig. 2.

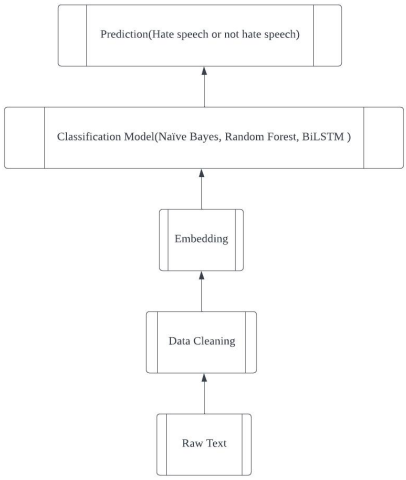


Fig. 1. Flow of classification for Naïve Bayes, Random Forest, and BiLSTM (Photo/Picture credit: Original).

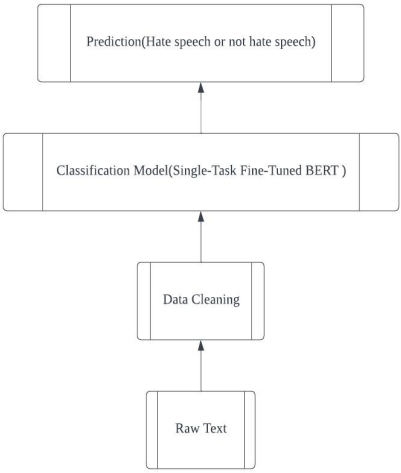


Fig. 2. Flow of classification for fine-tuned BERT (Photo/Picture credit: Original).

Naïve Bayes Classifier. The initial baseline model that was experimented with is a probabilistic learning method based on Bayes theorem, employing a multinomial naïve Bayes classifier. This classifier operates under the naïve assumption, presuming the independence of distinct features. Bag Of Words (BOW) was used as the word embedding method, and it refers to an unordered set of unique words that appeared in the entire dataset with their frequency of appearance kept. The vectorization process

involves transforming text documents to a matrix of word counts. The goal of this model is to get the correct class (hate speech or not hate speech) by choosing the class that has the maximum posterior probability given a comment from the dataset. This requires the naïve Bayes assumption to be made to simplify the calculation process, and the assumption is that all the features or the words in a comment are independent of each other. However, intuitively this is rarely true in real comments as words located at the end in a sequence can be highly dependent to words that are at the beginning. Furthermore, Laplace smoothing is applied during the calculation process to avoid getting a probability of 0 in the case of not having a training comment with a specific word associated with a label.

Random Forest Classifier. The random forest model used consists of 100 decision trees, and each of them is exposed to a different subset of the training data during the training process. Each tree generates a predicted class based on the input, and the most popular predicted class is taken as the prediction of the random forest classifier. The embedding method used for this classifier is BOW, which is the same as the Naïve Bayes classifier. During training, the nodes of a tree were expanded until all the leaves are completely pure or all leaves contain less than 2 samples. For building decision trees in the training process, Gini impurity was used as a measurement to determine the optimal split at each node.

BiLSTM Classifier. The LSTM architecture is a type of RNN that is capable of processing sequences of data, effectively utilizing previous inputs to influence the current input and output. Unlike a vanilla RNN, LSTMs excel at learning long-term dependencies in a sequence by using multiple "gates" that contribute to remembering information for a long period of time and control the flow of information into and out of the LSTM cell. The different gates work together to enable the model to choose what to store and what to forget at each cell during the training process. The sigmoid activation function is applied to all the different gates (i.e., forget gates, input gate, output gate). Bidirectional LSTM (BiLSTM) is a modification to the LSTM architecture in which two LSTMs are fed with the same sequence of data in two different directions, allowing the network to produce a more meaningful output by leveraging future contexts for each word.

In this study, the word embedding was initialized by transforming each text in a sequence to the index of the token in a dictionary of the vocabulary. The initialized word embedding was the input to the embedding layer which enabled the embeddings for each word to be learned during training. The resulting two vectors from BiLSTM of 128 nodes were concatenated together and the combined vector was fed to a dense layer with 32 nodes along with relu activation. This was followed by the dropout of 0.5, and another dense layer with 2 nodes and softmax activation.

BERT Classifier. BERT is a large pre-trained language model that is viewed as a base layer of knowledge as it had been pre-trained on an absurd amount of text with the two pre training objective (masked language modeling and next sentence prediction) [9]. It is a deep bidirectional model that is able to capture both the left and the right context in a sequence. BERT-base model was used in this experiment and it contains 12 transformer blocks, 12 attention head, and 768 hidden units. To fine tune BERT for sequence classification, this paper used a common fine tuning technique in

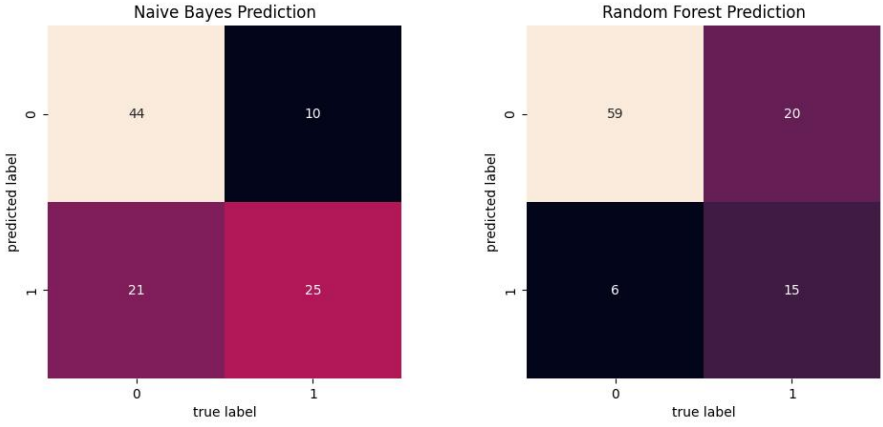
[9, 14] by feeding the [CLS] representation which is a special classification embedding that represents the entire sequence to an output dense layer with softmax activation. All the pre-trained model weights and the weights in the output layer will be modified during the training process.

3 Results and Discussion

This paper experimented with 4 different machine learning models, and Table 1 demonstrates the performance of all 4 different models on the testing dataset. The single-task fine-tuned BERT achieved the best performance on all four metrics in terms of accuracy, precision, recall and F1 Score among the 4 different models. The machine learning methods Naïve Bayes and Random Forest both resulted in an impressive accuracy of 0.69 and 0.70. Surprisingly, the accuracy, precision, recall, and F1 score for both Naïve Bayes and BiLSTM are very similar, and this can also be observed in Fig. 3 which shows the confusion matrices for the 2 models.

Table 1. Models Evaluation Results.

Model	Accuracy	Precision	Recall	F1 Score
Naïve Bayes	0.690	0.543	0.714	0.617
Random Forest	0.740	0.714	0.428	0.535
BiLSTM	0.680	0.533	0.686	0.600
BERT	**0.840**	**0.756**	**0.800**	**0.777**



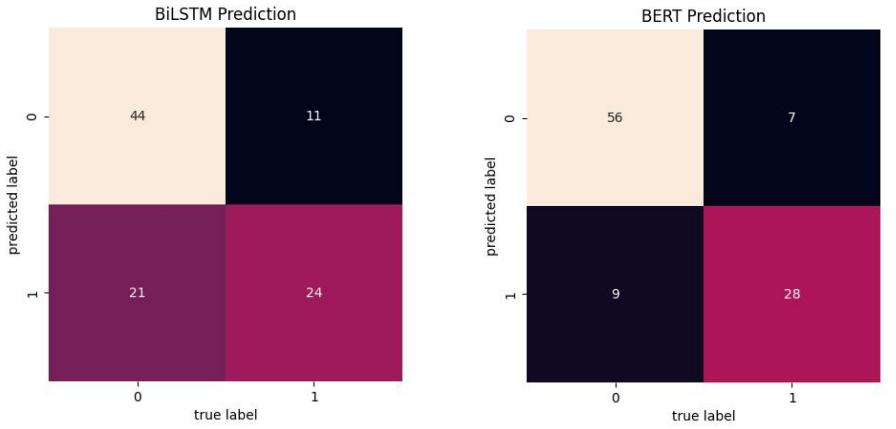


Fig. 3. Confusion Matrices for different models: Naïve Bayes, Random Forest BiLSTM, BERT where 1 means hate speech and 0 means not hate speech (Photo/Picture credit: Original).

The recall metric for Random Forest model is below 0.5, which is the lowest among all models. This indicates that there is a high number of false negatives in the prediction of the Random Forest classifier, and this can be observed from the confusion matrix for Random Forest in Fig. 3 with 20 hate speech being predicted as not hate speech. A plausible explanation for this outcome can be attributed to the utilization of an imbalanced dataset for model training. The binary version of ETHOS dataset has 57% of the data as not hate speech and 43% as hate speech. This slight imbalance could lead to a bias in the trained model towards the majority class as Random Forest is sensitive to imbalanced training data.

The confusion matrices in Fig. 3 show that Naïve Bayes and BiLSTM tend to have a higher number of false positive cases, namely the comments that are not hate speeches were predicted as hate speeches. The 21 false positive cases for both models are also very similar as 17 of them are the same comments from the testing dataset. One of the common characteristics that is shared between many of the 17 comments is that offensive language and sometimes swear words were present for praising purposes. This kind of comments are challenging to classify because they seem to be hated speeches due to the use of offensive language, but in reality, the offensive language was just used by the person that wrote the comment to better express the excitement and admiration. This could indicate that the trained Naïve Bayes and BiLSTM failed to capture the underlying meaning of this kind of sentence, but instead they judged the sentence by the appearance of certain words. Moreover, another characteristic of the false positive cases is the appearance of the words that describe skin color. For example, the comment “b bl bla blac black or w wh whi w” was predicted as hate speech by all four models, yet it is in fact not a hate speech as it does not express any kind of hatred towards a specific group of people. The words ‘black and ‘white’ as well as swear words appear in hate speech very frequently, but they are not always tied to hate speech and can appear in daily conversations and even compliments to others. This inherent complexity underscores the challenging nature

of hate speech detection, where a deep understanding of the contextual meaning within text sequences is crucial for achieving highly effective classification models. BERT, being the best performing model, also made a relatively small number of wrong predictions in challenging comments such as the one stated before. A larger dataset could contribute to solving this issue as models can have more opportunities to learn the more complex relationships and patterns from the diverse examples.

4 Conclusion

This study compared the effectiveness of several machine learning algorithms on hate speech detection. This paper formatted the problem as a sequence classification task with binary labels, where the labels are hate speech and not hate speech. Four different models, namely Naive Bayes, Random Forest, BiLSTM, and BERT, with different kinds of embedding methods were trained(fine-tuned) on the binary version of the ETHOS dataset which contains recent comments on YouTube and Reddit. After thorough experimentation, this study shows that the single-task fine-tuned BERT significantly outperformed all the other 3 models, and achieved the highest score in accuracy, precision, recall, and F1 score. Surprisingly, the efficient probabilistic classifier Naive Bayes is also demonstrated to be effective and achieved the second-best performance on all 4 metrics. In the future, different kinds of embedding methods and model architectures should be explored on a larger dataset related to a variety of social media platforms.

References

1. Binny, M., Ritam, D., Pawan, G., Animesh, M.: Spread of hate speech in online social media. Proceedings of the 10th ACM Conference on web science, 173-182 (2019).
2. Oana, Ș., Diana-Maria, B.: Hate Speech in Social Media and Its Effects on the LGBT Community: A Review of the Current Research. Romanian Journal of Communications and Public Relations, vol. 23, no. 1, 47-55 (2021).
3. Alice, T., Eugenia, N., Annalina, S., Lara, F.: Thirty years of research into hate speech: topics of interest and their evolution. Scientometrics, vol. 126, 157-179 (2021).
4. Lopamudra, D., Sanjay, C., Anuraag, B., Beepa B., Sweta T.: Sentiment Analysis of Review Datasets using Naïve Bayes and K-NN Classifier. arXiv.org (2016).
5. Haihua, C., Lei, W., Jiangping, C., Wei, L., Junhua, D.: A comparative study of automated legal text classification using random forests and deep learning. Information Processing & Management, vol. 59, no. 2, 102798 (2022).
6. Lishuang, L., Liuke, J., Zhenchao, J., Dingxin, S., Degen, H.: Biomedical named entity recognition based on extended Recurrent Neural Networks. IEEE International Conference on Bioinformatics and Biomedicine (2015).
7. Dan, L., Jiang, Q.: Text sentiment analysis based on long short-term memory. 2016 First IEEE International Conference on Computer Communication and the Internet, 471-475 (2016).

8. Ashish, V., Noam, S., Niki, P., Jakob, U., Llion, J., Aidan N. G., Lukasz, K., Illia, P.: Attention Is All You Need. Advances in Neural Information Processing Systems, 5998-6008，(2017).
9. Jacob, D., Ming-Wei, C., Kenton, L., Kristina, T.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv.org (2018).
10. Gaurav, R., Narinder, S. P., Sanjay, K. S., Sonali, A.: Hate speech detection using static BERT embeddings. Big Data Analytics (2021).
11. Abhishek, V., Hrushikesh, P., Amol, G., Shubham, S., Raviraj, J.: Hate and Offensive Speech Detection in Hindi and Marathi. arXiv.org (2021).
12. David, R., Ziqi, Z., Jonathan, T.: Hate Speech Detection on Twitter: Feature Engineering v.s. Feature Selection. The Semantic Web: ESWC 2018 Satellite Events (2018).
13. Ioannis, M., Zoe, C., Stamatis, K., Grigorios, T.: ETHOS: an Online Hate Speech Detection Dataset. arXiv.org (2021).
14. Chi, S., Xipeng, Q., Yige, X., Xuanjing, H.: How to Fine-Tune BERT for Text Classification?. Chinese Computational Linguistics (2019).