Applying Transfer Learning using BERT-based models for Hate Speech Detection

Sakshi Kalra[a], Kalit Naresh Inani[a], Yashvardhan Sharma[a] and Gajendra Singh Chauhan[b]

[a]*Department of Computer Science and Information Systems, Birla Institute of Technology and Science Pilani, Pilani Campus, Rajasthan, India*

[b]*Department of Humanities and Social Sciences, Birla Institute of Technology and Science Pilani, Pilani Campus, Rajasthan, India*

Abstract

Hateful and Offensive speech is rising along with social media. This issue has motivated researchers to devise novel approaches which perform better than the traditional algorithms. This paper presents the methods adopted by the BITS Pilani team for Subtask 1A of the Hate Speech and Offensive Content Identification in English and Indo-Aryan Language task proposed by the Forum of Information Retrieval Evaluation in 2021. We have used data augmentation to make the models generalize better. We have experimented with different feature extraction techniques along with machine learning algorithms. But, fine-tuning the pre-trained BERT-based models using transfer learning gave us the best results for all the given languages on the test set. We got the highest Macro-F1 of 0.7993 for the English Language, 0.7612 for the Hindi Language, and 0.8306 for the Marathi Language using the pre-trained BERT-based models.

Keywords

offensive language detection, hate speech, label classification, BERT-variants, HASOC

1. Introduction

Over the past years, the user base of social media platforms and online forums has grown exponentially. Every day around 500 million tweets are generated. People use these platforms to express their views and share other relevant information. But, since people come from different backgrounds, sometimes they might get hit by hateful, offensive, and objectionable speech. These issues arise due to the platform's anonymity allowing people to use racist, fanatic, and offensive terms in their speech. If unchecked, this poses a significant threat to society.

As a consequence, social media platforms need to monitor all their user posts. But, detecting and removing offensive speech by humans would require tremendous effort. Thus, a need arises to automate this task using modern machine learning and natural language processing algorithms. Toxic speech has two parts: hate and offensive speech. According to the UN, hate speech could be defined as "any communication in speech, writing or behavior, that attacks or

Forum for Information Retrieval Evaluation, December 13-17, 2021, India

p20180437@pilani.bits-pilani.ac.in (S. Kalra); f20180207@pilani.bits-pilani.ac.in (K. N. Inani); yash@pilani.bits-pilani.ac.in (Y. Sharma); gsc@pilani.bits-pilani.ac.in (G. S. Chauhan)

https://www.bits-pilani.ac.in/pilani/yash/profile (Y. Sharma); https://www.bits-pilani.ac.in/pilani/yash/profile (G. S. Chauhan)

uses pejorative or discriminatory language concerning a person or a group based on who they are, in other words, based on their religion, ethnicity, nationality, race, color, descent, gender or other identity factors" while offensive speech could be defined as "causes someone to feel resentful, upset, or annoyed." Finding common patterns in such text as tricky as people from different geographical and cultural backgrounds use it differently.

Online communities, social media enterprises, and technology companies are investing heavily and encouraging research in this area by organizing tasks and workshops. One such community is FIRE, actively managing the HASOC tasks since 2019[1]. This paper contains details regarding - Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages (SubTask A). The task is aimed at classifying a user tweet as either hate and offensive or not. We show the superiority of applying transfer learning on pre-trained BERT models over traditional machine learning algorithms.

1.1. Key Contributions

1. This paper shows the application of transfer learning by using pre-trained BERT models for hate and offensive speech detection.
2. The dataset used for the task was obtained by joining the data provided by the HASOC team for the year 2021 with the past two years' data. This would make our model generalize better.
3. Before feeding the data into our models, appropriate text processing techniques like lemmatization, removing stop words, removing mentions, URLs, etc. have been performed.
4. For feature extraction, techniques like TF-IDF weightings as well as word embeddings representations are used. These extracted features were then fed into machine learning algorithms namely logistic regression, random forest, and support vector classifier.
5. We have compared the BERT-based models with other machine learning models which use traditional natural language processing approaches for feature extraction. From the comparative study, it can be concluded that fine-tuned BERT-based models are the best suited for the above task.

2. Related Work

Various machine learning and deep learning approaches have been tested for automated hate and offensive speech detection[2]. The majority of the traditional machine learning approaches use feature extraction from speech text like a bag of words, n-grams, lexical and linguistic features[3]. Recently, word embedding methods have also been proposed for such tasks[4]. But these approaches fail to capture the entire context of the speech. Today, deep learning approaches[5] are gaining popularity in text classification, sentiment analysis, language modeling, machine translation, and many more. Some of these approaches are Convolutional Neural Networks(CNNs)[6], Recurrent Neural Networks(RNNs) [7], Long Short-Term Memory(LSTMs)[8], and the most recent is a transformer-based architecture [9] namely Bidirectional Encoder Representations(BERT)[10].

In [11] the authors provide a concise outline of the three shared tasks raised at the PAN 2021 lab on computerized text forensics and stylometry aided at the CLEF conference. The undertakings include authorship confirmation across domains, creator profiling for discourse spreaders, and style change disclosure for multi-writer documents. To a limited extent, they continue and advance past shared tasks, with the general objective of promoting state-of-the-art, accommodating an objective evaluation on recently created benchmark datasets. In [12] author uses various machine learning algorithms based on regression and classification as per the task requirement is to classify the hate speech and offensive words in the code-mixed language. Feature extraction is done using TF-IDF and n-grams based models on the dataset collected from the HASOC 2020 task, consisting of the Malayalam and Tamil Languages records, and got the F1-score of 0.77 for the Malayalam and 0.87 for the Tamil language. One more work was reported [13]for detecting the hate speech words on Twitter. The deep convolutional neural network model has been incorporated along with the GloVe embedding vectors to understand semantics. The results show that their model outperformed the existing models by achieving a F1-score of 0.92.

3. Proposed Techniques and Algorithms

The paper describes various approaches and draws out a comparison between them. The first approach extracts N-grams features from the preprocessed text and are weighted according to TF-IDF values. Then, models using machine learning algorithms are trained upon these features. The second approach uses word embeddings for numeric word representations, and models implementing machine learning algorithms are trained similarly to the previous method. Finally, a pre-trained BERT model is employed for training. This BERT model with twelve layers is trained on a large corpus of English data in a self-supervised way. This means it is trained on the raw texts only, with no humans labeling them in any way with an automatic process to generate inputs and labels from those texts. As a result, it tends to provide a better generalization than models trained from scratch. For the model adaptation for our task, it is fine-tuned and trained upon the dataset provided. Along with this, variations of BERT models like RoBERTa [14] and DistilBERT [15] have been used. Multilingual models are used for training and to classify the data from the languages such as Hindi and Marathi. Figure 1 describes the complete methodology we adopted for our experiments. The code is available in the github repository. [1]

4. Dataset

We performed data augmentation to make our models generalize better on new data. Thus, the dataset used for training was created by combining the organizers' datasets for HASOC 2021[16, 17], 2020, and 2019. Due to the unavailability of datasets from previous years, only data provided for HASOC 2021 was used for the Marathi language. The combined dataset consists of labeled tweets with the following classes:

[1]https://github.com/Kalit31/HASOC-2021

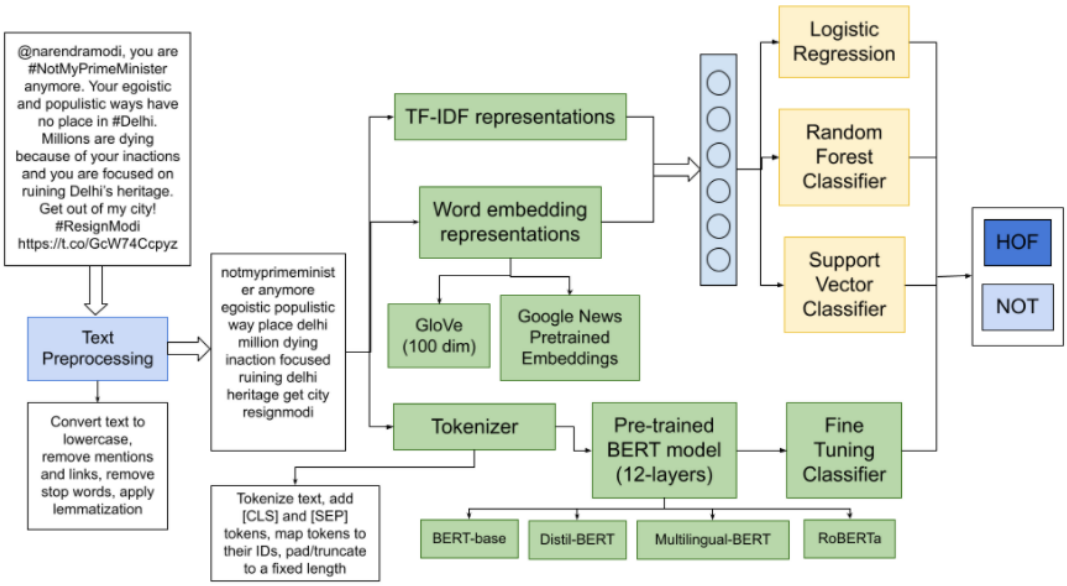

Figure 1: Flowchart of our methodology and techniques

Table 1
Dataset Statistics

Language	Train Data	Test Data
English	14556	1281
Hindi	13540	1532
Marathi	1874	625

- Non-Hate-Offensive(NOT) - Tweets with this label does not contain any Hate speech, profane, offensive content.
- Hate and Offensive(HOF) - Tweets with this label includes Hate, offensive, and profane content.

Along with this, each tweet is labeled with a HASOC ID provided by the organizers. Table 1 shows the statistics of the dataset used after concatenation of data from previous years.

5. Experimental Work

5.1. Machine Learning Algorithms using TF-IDF Representations

Firstly, the given tweets are preprocessed before the feature extraction part. For the English language, we convert the tweets to lowercase and remove the extra spaces, URLs, Twitter mentions, stopwords, and tokenize them using the functions available in the 'NLTK' package.

Table 2
An overview of hyper parameter setting used

Hyperparameter	Description
Optimizer	Adam
Learning Rate	2e-5
Number of Epochs	8
Batch Size	32

Preprocessing is done for the Hindi and Marathi languages using the 'iNLTK' package [18] similarly. Then, n-gram TF-IDF features are extracted. Here, n is a variable that ranges from one to three. Here, we use three machine learning algorithms: Logistic Regression, Support Vector Classifier, and Random Forest Classifier available in the 'scikit-learn'package [2] [19]. While training, a 5-fold grid search is performed to find the best set of hyperparameters. Logistic Regression gave the best performance for the English language, and the Support Vector Classifier worked well for the other two languages.

5.2. Machine Learning Algorithms using Word Embedding Representations

Here, we have experimented with two types of word embedding representations. One of them is the GloVe(100 D) embeddings. GloVe [20] is an unsupervised learning algorithm for obtaining vector representations for words. Model Training is performed on aggregated global word-to-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. The other approach is using Google News pre-trained Word2Vec model. Next, the mean of all the individual embeddings of words in a tweet is taken to get the numeric vector representation of the text. Then, we trained each model, using machine learning algorithms discussed above, on these vector features. A 5-fold grid search is used to get the best set of hyperparameters. Support Vector Classifier gave the best performance for all the languages.

5.3. BERT and its Variations

We experimented with various pre-trained transformer-based models provided by the Hugging Face Package [21]. We experimented with the following models for the English language: bert-base-uncased, roberta-base and distilbert-base-uncased. The following models were tested for Hindi and Marathi: bert-base-multilingual-cased, distilbert-base-multilingual-cased and indic-bert. The tweets were tokenized using transformer specific tokenizers. Then, a transformer specific model was used for sequence classification. Hyperparameter tuning is done to get the best results. Table 2 lists all the hyperparameters used while model training.

[2]https://scikit-learn.org/stable/

Table 3
Evaluation metrics for English using n-gram TF-IDF based models

Model	Macro-F1	Accuracy
Logistic Regression	0.7477	77.28%
Random Forest Classifier	0.7371	75.09%
Support Vector Classifier	0.7443	76.73%

Table 4
Evaluation metrics for English using word embedding representation models

Model	Macro-F1	Accuracy
LR + GloVe Embeddings	0.6816	68.93%
SVC + GloVe Embeddings	0.6963	71.19%
LR + Google News Embeddings	0.7097	71.97%
SVC + Google News Embeddings	0.7323	74.39%

Table 5
Evaluation metrics for English by finetuning BERT-based models

Model	Macro-F1	Accuracy
RoBERTa-base	0.7874	81.10%
BERT-base	0.7993	81.34%
DistilBERT-base	0.8065	82.51%

Table 6
Evaluation metrics for Hindi subtask

Model	Macro-F1	Accuracy
SVC + n-gram TF-IDF	0.7013	76.57%
BERT multilingual base	0.7505	79.76%
DistilBERT multilingual base	0.7612	80.35%

6. Results and Evaluations

All model performances are evaluated on the basis of Macro F1 and Accuracy. In the first approach using TF-IDF word representations, the Support Vector Classifier model performed the best compared to the other two algorithms. Similarly, for the model using word embeddings, the Support Vector Classifier performed well. We could not use the word embedding model for Hindi and Marathi due to library limitations. For all the languages, the pre-trained BERT-based models performed better than the feature extraction approaches. Though, all BERT-based variations seemed to give similar performance. DistilBERT multilingual performed slightly better than the BERT-multilingual base for the Hindi language, while it was the opposite case for the Marathi language. The results are tabulated in Tables 3, 4, 5, 6 and 7.

Table 7
Evaluation metrics for Marathi subtask

Model	Macro-F1	Accuracy
SVC + n-gram TF-IDF	0.7249	76.48%
IndicBERT	0.7505	80.48%
DistilBERT multilingual base	0.8072	83.04%
BERT multilingual base	0.8305	85.12%

7. Conclusions and Future Work

We can see from the above results that the pre-trained BERT models are better and able to capture the context of a given sentence and thus provide better representation for learning. Therefore, the transfer learning approach on pre-trained BERT models is better suited for identifying hate and offensive speech than the traditional feature extraction approaches. For the future scope, the performance for the Indian languages, namely - Hindi and Marathi, could be improved by using better word tokenization with specific tokens for Indian language words. The low performance on the Marathi language may be due to the limited data compared to the other two languages. Thus, models can be trained on a larger corpus in the future. Moreover, deeper transformer architectures may be tried out in the future.

References

[1] P. Mehta, T. Mandl, P. Majumder, S. Gangopadhyay, Report on the fire 2020 evaluation initiative, in: ACM SIGIR Forum, volume 55, ACM New York, NY, USA, 2021, pp. 1–11.

[2] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 11, 2017.

[3] A. Gaydhani, V. Doma, S. Kendre, L. Bhagwat, Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach, arXiv preprint arXiv:1809.08651 (2018).

[4] R. Kshirsagar, T. Cukuvac, K. McKeown, S. McGregor, Predictive embeddings for hate speech detection on twitter, arXiv preprint arXiv:1809.10644 (2018).

[5] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, in: Proceedings of the 26th international conference on World Wide Web companion, 2017, pp. 759–760.

[6] Z. Zhang, L. Luo, Hate speech detection: A solved problem? the challenging case of long tail on twitter, Semantic Web 10 (2019) 925–945.

[7] G. K. Pitsilis, H. Ramampiaro, H. Langseth, Effective hate-speech detection in twitter data using recurrent neural networks, Applied Intelligence 48 (2018) 4730–4742.

[8] A. Bisht, A. Singh, H. Bhadauria, J. Virmani, et al., Detection of hate speech and offensive language in twitter data using lstm model, in: Recent trends in image and signal processing in computer vision, Springer, 2020, pp. 243–264.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
[10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[11] J. Bevendorff, B. Chulvi, G. L. D. L. Peña Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, et al., Overview of pan 2021: Authorship verification, profiling hate speech spreaders on twitter, and style change detection, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2021, pp. 419–431.
[12] V. Pathak, M. Joshi, P. Joshi, M. Mundada, T. Joshi, Kbcnmujal@ hasoc-dravidian-codemix-fire2020: Using machine learning for detection of hate speech and offensive code-mixed social media text, arXiv preprint arXiv:2102.09866 (2021).
[13] P. K. Roy, A. K. Tripathy, T. K. Das, X.-Z. Gao, A framework for hate speech detection using deep convolutional neural network, IEEE Access 8 (2020) 204951–204962.
[14] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
[15] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).
[16] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schäfer, T. Ranasinghe, M. Zampieri, D. Nandini, A. K. Jaiswal, Overview of the HASOC subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021. URL: http://ceur-ws.org/.
[17] S. Gaikwad, T. Ranasinghe, M. Zampieri, C. M. Homan, Cross-lingual offensive language identification for low resource languages: The case of marathi, in: Proceedings of RANLP, 2021.
[18] G. Arora, inltk: Natural language toolkit for indic languages, arXiv preprint arXiv:2009.12534 (2020).
[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830.
[20] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
[21] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Huggingface's transformers: State-of-the-art natural language processing, arXiv preprint arXiv:1910.03771 (2019).

A. Online Resources

The implementation of different pre-trained BERT-models are available at

- Huggingface.