HASOCOne@FIRE-HASOC2020: Using BERT and Multilingual BERT models for Hate Speech Detection

Suman Dowlagar[a], Radhika Mamidi[a]

[a]*International Institute of Information Technology - Hyderabad (IIIT-Hyderabad), Gachibowli, Hyderabad, Telangana, India, 500032*

Abstract

Hateful and Toxic content has become a significant concern in today's world due to an exponential rise in social media. The increase in hate speech and harmful content motivated researchers to dedicate substantial efforts to the challenging direction of hateful content identification. In this task, we propose an approach to automatically classify hate speech and offensive content. We have used the datasets obtained from FIRE 2019 and 2020 shared tasks. We perform experiments by taking advantage of transfer learning models. We observed that the pre-trained BERT model and the multilingual-BERT model gave the best results. The code is made publically available at https://github.com/suman101112/hasoc-fire-2020

Keywords

Hate speech, offensive content, label classification, transfer learning, BERT

CEUR-WS.org/Vol-2826/T2-11.pdf

1. Introduction

Nowadays, people are frequently using social media platforms to communicate their opinions and share information. Although the communication among users can lead to constructive conversations, the people have been increasingly hit by hateful and offensive content due to these platforms' anonymity features. It has become a significant issue. The threat of abuse and harassment made many people stop expressing themselves.

According to the Cambridge dictionary, Hate speech and offensive content is defined as,

- To harass and cause lasting pain by attacking something uniquely dear to the target.
- To use words that are considered insulting by most people.

The main obstacle with hate speech is, it is difficult to classify based on a single sentence because most of the hate speech has context attached to it, and it can morph into many different shapes depending on the context. Another obstacle is that humans cannot always agree on what can be classified as hate speech. Hence it is not very easy to create a universal machine learning algorithm that would detect it. Also, the datasets used to train models tend to "reflect the majority view of the people who collected or labeled the data".

To deal with the above scenarios and to encourage research on hate speech and offensive content, the NLP community organized several tasks and workshops such as Task 12: OffensEval 2: Multilingual Offensive content identification in Social Media text [1], OSATC4 shared task on offensive content

FIRE '20, Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India.

suman.dowlagar@research.iiit.ac.in (S. Dowlagar); radhika.mamidi@iiit.ac.in (R. Mamidi)

0000-0001-8336-195X (S. Dowlagar)

© 2020 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

[1] https://sites.google.com/site/offensevalsharedtask/

detection [2]. Similarly, the FIRE 2020's HASOC shared task was devoted to the Hate Speech and Offensive Content Identification in Indo-European Languages. This task aims to classify the given annotated tweets. This paper presents the state-of-the-art BERT transfer learning models for automated detection of hate speech and offensive content.

The paper is organized as follows. Section 2 provides related work on hate speech and offensive content detection. Section 3 describes the methodology used for this task. Section 4 presents the experimental setup and the performance of the model. Section 5 concludes our work.

2. Related Work

Machine learning and natural language processing approaches have made a breakthrough in detecting hate speech on web platforms. Many scientific studies have been dedicated to using Machine Learning (ML) [1, 2] and Deep Learning (DL) [3, 4] methods for automated hate speech and offensive content detection. The features used in traditional machine learning approaches are word-level and character-level n-grams, etc. 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, they necessitate a well-defined feature extraction approach. Nowadays, the neural network models apply text representation and deep learning approaches such as Convolutional Neural Networks (CNNs) [5], Bi-directional Long Short-Term Memory Networks (LSTMs) [6], and BERT [7] to improve the performance of hate speech and offensive content detection models.

3. Methodology

Here, we use the pre-trained BERT transformer model for hate speech and offensive content detection. Figure 1 depicts the abstract view of BERT model that is used for hate speech detection and offensive language identification. Bidirectional Encoder Representations from Transformers (BERT) is a transformer Encoder stack trained on the large English corpus. It has 2 models, $BERT_{base}$ and $BERT_{large}$. These model sizes have a large number of transformer layers. The $BERT_{base}$ version has 12 transformer layers and the $BERT_{large}$ has 24. These also have larger feed-forward networks with 768 and 1024 hidden representations, and attention heads are 12 and 16 for the respective models. Like the vanilla transformer model [8], BERT takes a sequence of words as input. Each layer applies self-attention, passes its results through a feed-forward network, and then hands it off to the next encoder. Embeddings from $BERT_{base}$ have 768 hidden units. The BERT configuration model takes a sequence of words/tokens at a maximum length of 512 and produces an encoded representation of dimensionality 768.

The pre-trained BERT models have a better word representation as they are trained on a large Wikipedia and book corpus. As the pre-trained BERT model is trained on generic corpora, we need to fine-tune the model for the downstream tasks. During fine-tuning, the pre-trained BERT model parameters are updated when trained on the labeled hate speech and offensive content dataset. When fine-tuned on the downstream sentence classification task, a very few changes are applied to the $BERT_{base}$ configuration. In this architecture, only the [CLS] (classification) token output provided by BERT is used. The [CLS] output is the output of the 12th transformer encoder with a dimensionality of 768. It is given as input to a fully connected neural network, and the softmax activation function

[2] http://edinburghnlp.inf.ed.ac.uk/workshops/OSACT4/

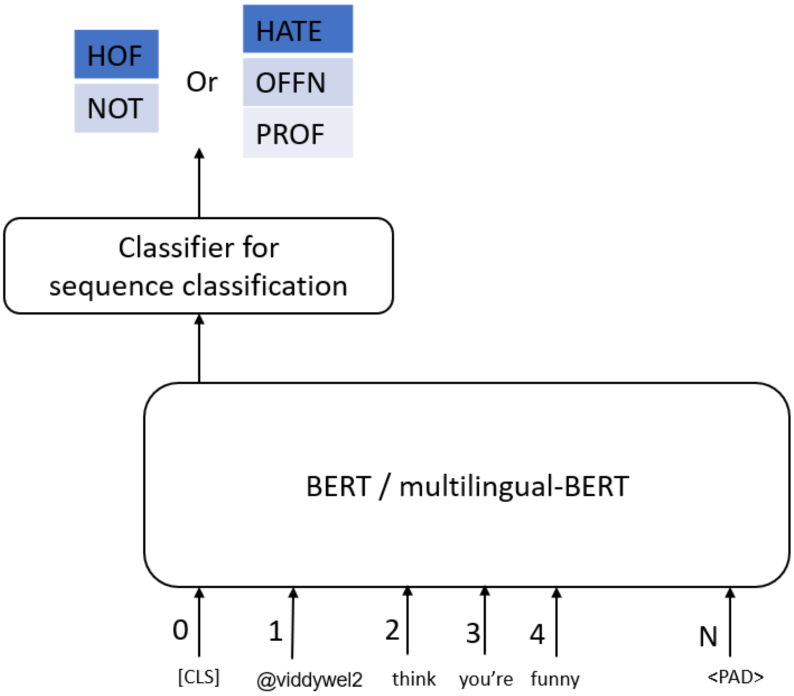


Figure 1: BERT model for sequence classification on Hate Speech Data.

is applied to the neural network to classify the given sentence. Thus, BERT learns to predict whether a tweet can be classified as a hate speech or offensive content. Apart from $BERT_{base}$ model, we used the pre-trained multilingual $BERT_{base}$ model, as our data consisted of German and Hindi multilingual languages. The multilingual BERT and vanilla BERT models' architecture is the same, but the pre-trained multilingual BERT model is trained on multilingual Wikipedia language sources.

4. Experiment

Initially, we introduce datasets used, the task description, and then review the BERT model's performance on hate speech and offensive content detection. We also include our implementation details and error analysis in the subsequent sections.

4.1. Dataset

We used the dataset provided by the organizers of HASOC FIRE-2020 [9] and FIRE-2019 [10]. The HASOC dataset was subsequently sampled from Twitter and partially from Facebook for *English*, *German*, and *Hindi* languages. The tweets were acquired using hashtags and keywords that contained offensive content. The statistics of FIRE 2020 and 2019 datasets are given in the Table 1.

4.2. Task description

The following tasks are in HASOC 2020.

Table 1
Data Statistics

Language	Train Sentences	Test Sentences
English (HASOC 2019)	5852	1153
German (HASOC 2019)	3819	850
Hindi (HASOC 2019)	4665	1318
English (HASOC 2020)	3708	814
German (HASOC 2020)	2373	526
Hindi (HASOC 2020)	2963	663

Sub-task A focuses on coarse-grained Hate speech detection in all three languages. The task is to classify tweets into two classes:

- (NOT) Non Hate-Offensive - Post does not contain any Hate speech, profane, offensive content.
- (HOF) Hate and Offensive - Post contains Hate, offensive, and profane content.

Sub-task B represents a fine-grained classification. Hate-speech and offensive posts from the sub-task A are further classified into three categories. The task is to classify the tweets into three classes:

- (HATE) Hate speech - Post contains Hate speech content.
- (OFFN) Offenive - Post contains offensive content such as insulting, degrading, dehumanizing and threatening.
- (PRFN) Profane - Post contains profane words. This typically concerns the usage of swearwords and cursing.

4.3. Implementation

For the implementation, we used the transformers library provided by HuggingFace [11]. The HuggingFace transformers package is a python library providing pre-trained and configurable transformer models useful for a variety of NLP tasks. It contains the pre-trained BERT and multilingual BERT, and other models suitable for downstream tasks. As the implementation environment, we use the PyTorch library that supports GPU processing. The BERT models were run on NVIDIA RTX 2070 graphics card with an 8 GB graphics card. We trained our classifier with a batch size of 64 for 5 to 10 epochs based on our experiments. The dropout is set to 0.1, and the Adam optimizer is used with a learning rate of 2e-5. We used the hugging face transformers pre-trained BERT tokenizer for tokenization. We used the BertForSequenceClassification module provided by the HuggingFace library during finetuning and sequence classification.

4.4. Baseline models

Here, we compared the BERT model with other machine learning algorithms.

4.4.1. SVM with TF_IDF text representation

We chose Support Vector Machines (SVM) for hate speech and offensive content detection. The tokenizer used is SentencePiece [12]. SentencePiece is a commonly used technique to segment words into a subword-level. In both cases, the vocabulary is initialized with all the individual characters in

Table 2
macro F1 and Accuracy on English Subtasks A and B

	Hate speech Detection		Offensive Content Identification	
Model	**macro F1**	**Accuracy**	**macro F1**	**Accuracy**
SVM	81.56%	81.57%	47.49%	76.78%
ELMO + SVM	82.43%	83.78%	49.62%	79.54%
BERT	88.33%	88.33%	54.44%	81.57%

Table 3
macro F1 and Accuracy on German Subtasks A and B

	Hate speech Detection		Offensive Content Identification	
Model	**macro F1**	**Accuracy**	**macro F1**	**Accuracy**
SVM	73.29%	79.27%	45.54%	77.94%
ELMO + SVM	71.73%	80.42%	45.94%	78.21%
multilingual-BERT	77.91%	82.51%	47.78%	80.42%

Table 4
macro F1 and Accuracy on Hindi Subtasks A and B

	Hate speech Detection		Offensive Content Identification	
Model	**macro F1**	**Accuracy**	**macro F1**	**Accuracy**
SVM	59.73%	70.13%	36.78%	72.39%
ELMO + SVM	60.91%	71.47%	39.89%	72.76%
multilingual-BERT	63.54%	74.96%	49.71%	73.15%

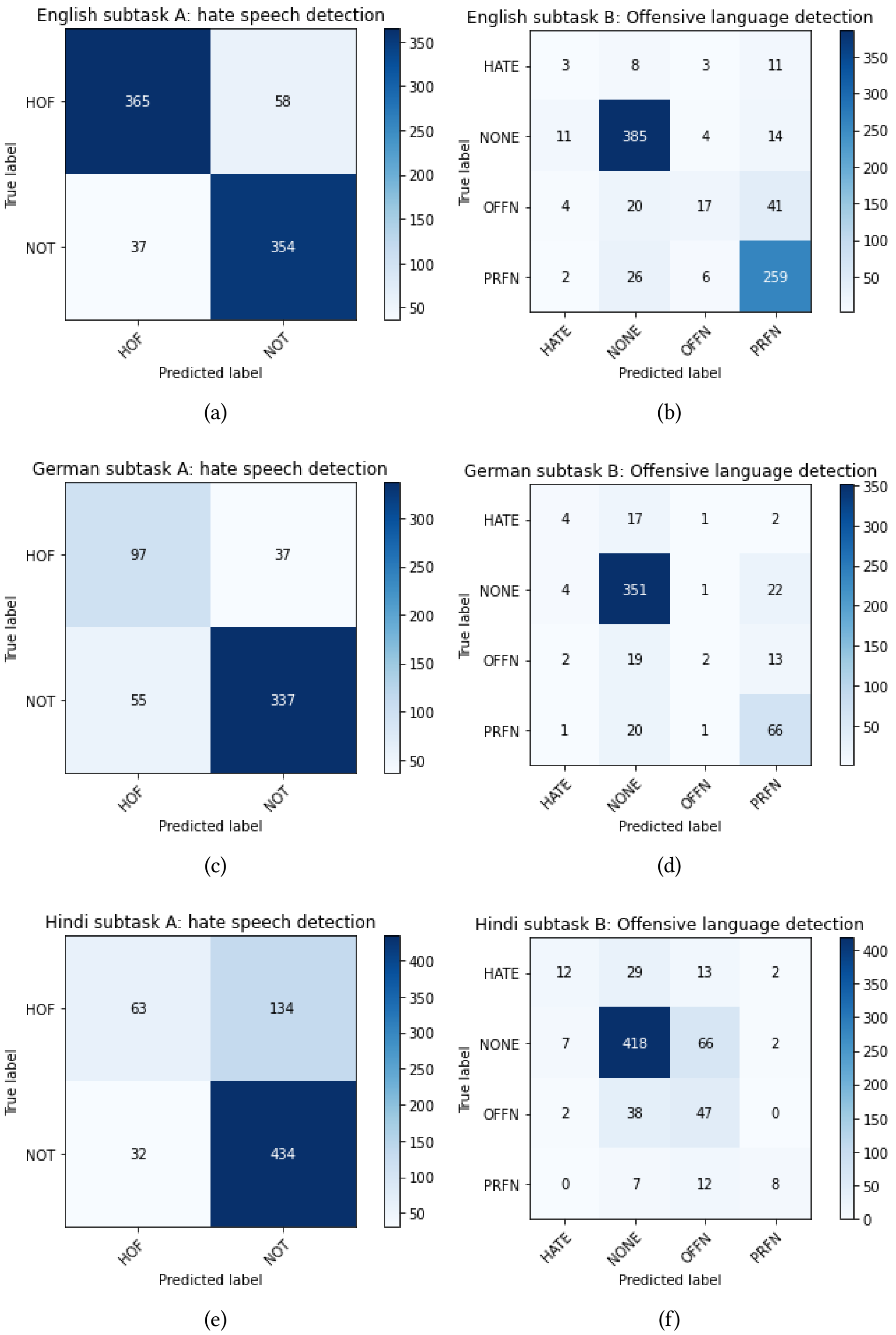
the language, and then the most frequent or likely combinations of the symbols are iteratively added to the vocabulary.

4.4.2. ELMO embeddings with SVM model

ELMO(Embeddings from Language Models) [13] deals with contextual embeddings. Contextual word-embeddings are born to capture the word meaning in its context. Instead of using a fixed embedding for each word, ELMO looks at the word's context, i.e., the word's entire sentence, before assigning embedding to the word. It uses a bi-LSTM trained on a specific task to be able to create those embeddings. We used the ELMO model present on tensorflow hub (https://tfhub.dev/google/elmo/2) to obtain the ELMO embeddings on the hate speech data for all the languages. After obtaining the embeddings, we take the mean of embeddings and apply an SVM classifier to classify the given sentence into hate speech or offensive content. We used the SentencePiece tokenizer.

5. Results

The results are tabulated in Tables 2, 3 and 4. We evaluated the performance of the method using macro F1 and accuracy. The BERT model performed well when compared to the other SVM with TF-IDF and ELMO text representations. Given all the languages and both the subtasks A and B, we have observed an increase of 1-2% in classification metrics for ELMO embeddings + SVM classifier compared to the baseline SVM classifier. However, BERT showed an increase of 5-7% in classification metrics compared to ELMO and SVM models. It shows the pre-trained BERT model's capability, which learnt better text representations from the generic data. The state of the art transformer architecture



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2: Confusion matrix on the given test data for the English, German and Hindi languages given subtask A: Hate Speech Detection and subtask B: Offensive Content Identification

used in the BERT model helped the model learn better parameter weights in hate speech and offensive content detection.

6. Error Analysis

The confusion matrix of BERT model for subtasks A and B for the english, german and hindi datasets is given in the Figure 2. For the binary classification, the best-performed model was for English subtask A. The binary classification for the Hindi model is not helpful. The model misclassified most of the hate-speech labels. It can be seen in subfigure 2(e). For offensive content evaluation, the model performed better on English subtask B. It correctly classified "NONE (not offensive)" and "PROF (profane)" but was unable to classify "HATE (hate speech)" and "OFFN (offensive)" and misunderstood most of them as "PROF". The multilingual-BERT model misclassified most of the hate speech and offensive content labels for the German and Hindi languages as "NONE" and didn't perform well on those datasets.

7. Conclusion and Future work

We used pre-trained bi-directional encoder representations using transformers (BERT) and multilingual-BERT for hate speech and offensive content detection for English, German, and Hindi languages. We compared the BERT with other machine learning and neural network classification methods. Our analysis showed that using the pre-trained BERT and multilingual BERT models and finetuning it for downstream hate-speech text classification tasks showed an increase in macro F1 score and accuracy metrics compared to traditional word-based machine learning approaches.

The given data has both hate speech and offensive content labeled for a given same sentence. It implies that both tasks are related. In such a scenario, we can use joint learning models to help obtain a strong relationship between the two tasks. Which, in turn, helps a deep joint classification model to understand the given datasets better.

References

[1] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, arXiv preprint arXiv:1703.04009 (2017).

[2] 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).

[3] B. Gambäck, U. K. Sikdar, Using convolutional neural networks to classify hate-speech, in: Proceedings of the first workshop on abusive language online, 2017, pp. 85–90.

[4] 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.

[5] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882 (2014).

[6] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[7] 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).

[8] 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.

[9] T. Mandl, S. Modha, G. K. Shahi, A. K. Jaiswal, D. Nandini, D. Patel, P. Majumder, J. Schäfer, Overview of the HASOC track at FIRE 2020: Hate Speech and Offensive Content Identification in Indo-European Languages), in: Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, CEUR, 2020.

[10] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, A. Patel, Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages, in: Proceedings of the 11th Forum for Information Retrieval Evaluation, 2019, pp. 14–17.

[11] 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 (2019) arXiv–1910.

[12] T. Kudo, J. Richardson, Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, arXiv preprint arXiv:1808.06226 (2018).

[13] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, arXiv preprint arXiv:1802.05365 (2018).