Hate Speech and Offensive Content Identification Based on Self-Attention

Yifan Xu[1], Hui Ning[1] and Yutong Sun[2]

[1]Harbin Engineering University, Harbin, China
[2]Heilongjiang Institute of Technology, Harbin, China

Abstract

With the development of the Internet, more and more people use the social medias to share their daily life. However, there are various problems existing in the online community. One of these problems is that some people would like to post hate speech and offensive contents. How to identify hate speech and offensive contents is a serious problem. "Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages" is a track which is committed to solve this problem. We used three different models consisting of SVM, CNN and BERT to do experiments about English texts. Among them, BERT has the best performance. Our team called QQQ get a Macro-averaged F1 score of 0.7374.

Keywords

Hate speech, BERT, Self-Attention

1. Introduction

With the development of the Internet, the number of people who surf the Internet is increasing. Nowadays, Tweet and Facebook are the most popular social medias. A lot of people post hate speech and offensive contents about race, nationality, religion, ethnicity and so on. In order to solve this problem we need a system that can automatically identify whether a text contains hate speech and offensive content.

The "Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages" (HASOC) is such a track which is committed to identify the hate speech and offensive content[1]. We participated in the HASOC 2021 and performed some experiments. HASOC offers 2 subtasks: subtask 1 and subtask 2. Subtask 1 consists of two tasks. Subtask 1A is identifying hate, offensive and profane content from the post. Subtask 1B is discrimination between hate, profane and offensive posts. Subtask 2 is Identification of Conversational Hate-Speech in Code-Mixed Languages (ICHCL). In this paper, we used the datasets of subtask 1A to do experiments with the models of SVM, CNN and BERT.

Section 2 will introduce the recent studies about the detection of hate speech. Section 3 will show the datasets released by HASOC. Then the methods of experiments will be introduced in the section 4. Section 5 reports the result of experiments. Section 6 will give a conclusion. References are at the end of this paper.

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

2. Related Work

A lot of methods have been used in the detection of hate speech and offensive content such as TF-IDF[2], Bag-of-words (BOW)[3] and Word embedding[4]. But the results of these methods seem not good. Using these methods, we can't distinguish the meaning of the words when we encounter the situation where the word has multiple meanings.

A number of recent studies show that the methods based on the deep learning have better performance than the traditional machine learning. Convolutional Neural Networks (CNN)[5] and Recurrent Neural Networks (RNN) such as Long Short-Term Memory (LSTM)[6] are the most common methods in the deep learning. These days, Self-Attention[7] is the most popular technology in tasks of Natural Language Processing (NLP). An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query(Q), keys(K), values(V), and output are all vectors. We compute the matrix of outputs as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{1}$$

Where d_k is the dimension of the query and key vectors. Based on these attention scores, each word is given a weighted vector representation that captures contextual information[7].

In this paper we choose to use the model of BERT[8] based on self-attention to perform the experiment.

3. Data

In this section, we will introduce the datasets and the method we process them.

3.1. Datasets

The training dataset is provided by HASOC which consists of ids, texts and labels of subtask 1A and 1B[9, 10, 11]. The labels of subtask 1A are "HOF" and "NOT". The labels of subtask 1B are "PRFN", "HATE" and "OFFN".

- **(NOT) Non Hate-Offensive:** This post does not contain any Hate speech, profane, offensive content.
- **(HOF) Hate and Offensive:** This post contains Hate, offensive, and profane content.
- **(HATE) Hate speech:** Posts under this class contain Hate speech content.
- **(OFFN) Offensive:** Posts under this class contain offensive content.
- **(PRFN) Profane:** These posts contain profane words.

The data is mainly from Tweet and Facebook in English, Hindi and German. In this paper, we choose to use the English dataset to perform the experiment. There are total 3790 texts in the English training dataset and 1268 texts in the test dataset. In addition to the dataset of HASOC 2021, for the subtask 1A, we also used the datasets of HASOC 2020 and HASOC 2019 which consist of 12305 texts in English in total. The size of test datasets is 1268 posts. The Table 1

Table 1
Size of datasets

Dataset	Number of texts
HASOC 2021	3790
HASOC 2019 and HASOC 2020	12305
Test data	1268

shows the datasets in detail. We need to use the training data to train our models. Then use the models to predict the labels of test data.

The data format in the dataset is presented in the Table 2.

Table 2
Data format

Id	Text	Task_1	Task_2
60c5d6bf5659ea5e55def461	Technically that's still turning back the clock, dick head https://t.co/jbKaPJmpt1	HOF	OFFN
60c5d6bf5659ea5e55def419	@krtoprak_yigit Soldier of Japan Who has dick head	HOF	OFFN
60c5d6bf5659ea5e55df0109	damn damson asked for some transparency and they made him deactivate	NOT	NONE

3.2. Datasets Segmentation

We combined the datasets of HASOC 2021, HASOC 2020 and HASOC 2019 into the training data and disrupted the order. At the same time, we delete the labels of task_2. After that, we divide the training data into a train set and a development set according to a 4:1 ratio.

The test set is released by HASOC including 1268 posts.

3.3. Preprocessing

To clean the data, we removed some useless strings such as URL, id and emoji. We also removed some special symbols. These strings may reduce the accuracy of experimental results. So we removed them in advance to improve the result of the experiment.

Table 3
Data after preprocessing

Id	Text	Task_1	Task_2
60c5d6bf5659ea5e55def461	Technically that's still turning back the clock dick head	HOF	OFFN
60c5d6bf5659ea5e55def419	Soldier of Japan Who has dick head	HOF	OFFN
60c5d6bf5659ea5e55df0109	damn damson asked for some transparency and they made him deactivate	NOT	NONE

4. Methods

In this section, we are going to introduce the methods that we used in the experiments. The methods are SVM, CNN and BERT.

4.1. Support Vector Machine

Support vector machine[12] is a binary classification model whose basic model is a linear classifier defined by maximizing the interval on the feature space, which distinguishes it from a perceptron. SVM also includes kernel tricks, which make it essentially nonlinear classifier. The learning algorithm of SVM is the optimization algorithm for solving convex quadratic programming.

4.2. Convolutional Neural Networks

CNNs are often used in image recognition systems. Besides that,CNNs have also been explored for natural language processing. CNN models are effective for various NLP problems and achieved excellent results in semantic parsing, sentence modeling, search query retrieval, classification, prediction and other traditional NLP tasks[13].

A convolutional neural network consists of an input layer, hidden layers and an output layer. In any feed-forward neural network, all of the middle layers are called hidden because their outputs and inputs are masked by the activation function and final convolution. In this paper, the input layer is the embedding layer. The embedding layer encodes the word used in the comments. We used fastText embedding as the embedding layer. After feeding the embedded comment to the CNN layer, we used four layers of convolution and one layer of max-pooling.Finally, we used the flatten layer followed by a dense layer. At the dense layer, sigmoid and softmax are activation functions which is good at binary class problems.In the hidden layers, activation function is the Rectified Linear Unit (ReLU).

4.3. BERT

BERT is a pre-training model based on self-attention which has good performance in many NLP tasks[14]. It evolved from the transformer model which is a kind of seq2seq model consisting of Encoder and Decoder.

In our experiment, we used the pre-trained models of ”BERT-Base, Uncased” released by Google which has 12-layer, 768-hidden, 12-heads, 110M parameters. We set max_seq_length=200, train_batch_size=32, learning_rate=2e-5 and num_train_epochs=3.0. In our task of identifying the hate speech, we only need to classify the text which is usually less than 200 words. We can regard the text as a sentence. We classify a sentence as follow:

Input = damn damson asked for some transparency and they made him deactivate
Label = NOT

5. Experiments

In this section, we will show our results of the experiments. We choose SVM and CNN as the baseline. We wiil find how much the Bert model can improve the experimental effect. And will there be differences between the data with preprocessing and without preprocessing in the results.

5.1. Evaluation Measures

In this paper, we used **Macro-averaged F1 score** as the evaluation measure.Here is the definition of the evaluation:

Condition positive (P):the number of real positive cases in the data.

Condition negative (N):the number of real negative cases in the data.

There are true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) in the result of binary classification. Precision and recall are then defined as:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Macro_P = \frac{\sum Precision}{n} \quad Macro_R = \frac{\sum Recall}{n} \tag{3}$$

$$Macro_F = \frac{2 \times Macro_p \times Macro_R}{Macro_p + Macro_R} \tag{4}$$

5.2. Result

We used SVM, CNN and BERT to perform the experiment. The evaluation measure is **Macro-averaged F1**. The result presents in the follow table.

Table 4
Classification result

	With preprocessing	Without preprocessing
SVM	0.6042	0.5871
CNN	0.6744	0.6657
BERT	0.7374	0.7216

In the table, we see that CNN has a slightly better performance than SVM in this task while BERT has a significant improvement compared with the two models mentioned above. 0.7374 is our best submission result which we named it QQQ_submission on the website of HASOC. As expected, the preprocessing of texts can also improve the evaluation of the result.

6. Conclusion

In recent years, the spread of hate speech has become more widespread. It has become a serious problem in the social medias. In this paper, we make a little contribution to this problem. We used the models of SVM, CNN and BERT to classify the hate speech. Among them, BERT has the best performance whose Macro-averaged F1 score is 0.7374.

Unfortunately, the training data is a little bit small. We may improve the model if we have more training data. Besides that, the number of two kinds of label is not equal.The gap between them is a bit big. This also has a bad effect on the experiment. In the future work, we will try some improved models based on BERT such as ERNIE[15] and RoBERTa[16]. We will also search for more data to enhance the training data.

References

[1] S. Modha, T. Mandl, G. K. Shahi, H. Madhu, S. Satapara, T. Ranasinghe, M. Zampieri, Overview of the HASOC Subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages and Conversational Hate Speech, in: FIRE 2021: Forum for Information Retrieval Evaluation, Virtual Event, 13th-17th December 2021, ACM, 2021.

[2] S. Agarwal, A. Sureka, Characterizing linguistic attributes for automatic classification of intent based racist/radicalized posts on tumblr micro-blogging website (2017).

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

[4] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, International World Wide Web Conferences Steering Committee (2017).

[5] A. Conneau, H. Schwenk, L. Barrault, Y. Lecun, Very deep convolutional networks for text classification, Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers (2017).

[6] D. Wei, B. Wang, G. Lin, D. Liu, Z. Dong, H. Liu, Y. Liu, Research on unstructured text data mining and fault classification based on rnn-lstm with malfunction inspection report, ENERGIES (2017) 1–22.

[7] 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.

[8] 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).

[9] 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/.

[10] 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.

[11] S. Satapara, S. Modha, T. Mandl, H. Madhu, P. Majumder, Overview of the HASOC Subtrack at FIRE 2021: Conversational Hate Speech Detection in Code-mixed language , in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[12] A. Wright, A. B. McCoy, S. Henkin, A. Kale, D. F. Sittig, Use of a support vector machine for categorizing free-text notes: assessment of accuracy across two institutions, Journal of the American Medical Informatics Association 20 (2013) 887–890.

[13] W. Wang, J. Gang, Application of convolutional neural network in natural language processing, in: 2018 International Conference on Information Systems and Computer Aided Education (ICISCAE), IEEE, 2018, pp. 64–70.

[14] M. Mozafari, R. Farahbakhsh, N. Crespi, Hate speech detection and racial bias mitigation in social media based on bert model, PloS one 15 (2020) e0237861.

[15] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, H. Wang, Ernie 2.0: A continual pre-training framework for language understanding, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 8968–8975.

[16] 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).