Simon@HASOC 2020: Detecting Hate Speech and Offensive Content in German Language with BERT and Ensembles

Qinyu Que[a], Ruijie Sun[b] and Shasha Xie[c]

[a]School of Information Science and Engineering, Yunnan University, Yunnan, P.R. China

Abstract

In this paper, we introduce the system for the Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC) 2020 Challenge, which is submitted by our team. We use a lot of social media in our daily life, but now social media is full of hate speech and offensive language, so the detection of hate speech and offensive language has become an essential task. The task is available in English, German, and Hindi, but there is a lot of work done in the English languages, with limited work reporting posts in Hindi and German, so we chose the German task to complete. The BERT-Ger model could not meet our requirements for semantic information characteristics, we modify the upper layer structure of BERT-Ger. Finally, our system wins second place in German subtask A and tenth in German subtask B.

Keywords

Hate Speech, Offensive, German, BERT

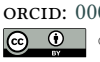
1. Introduction

With the rapid development of network information technology, we are more and more used to express our opinions on social media, such as Facebook, Twitter, etc., but there is a lot of hate speech and offensive language in these published contents. Hate speech is published in cyberspace and spread through the network media, aiming at groups with specific identities to carry out offensive and harmful speech [1]. The biggest difference between internet violence and the traditional behavior of defamation lies in the fact that the objects of hate speech on the Internet are specific groups with high identifiability [2]. Due to the changeable international situation, local conflicts still exist, and international problems such as multi-national debt crisis, refugee crisis, terrorist attack crisis, violent crime, and immigration are intertwined with each other, the hate speech on the Internet is growing day by day, and the thoughts of terror, violence and extremism in the network platform take the opportunity to penetrate, which has brought great negative impact on social security. The governance of network hate speech has gradually attracted people's attention. Nowadays, such social media companies are studying the identification of hate speech and offensive language, which is very difficult because some sentences containing neutral words are difficult to detect as hate words, but

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

EMAIL: 1309487642@qq.com (Q. Que)

ORCID: 0000-0001-6688-7896 (Q. Que); 0000-0003-2987-5743 (R. Sun); 0000-0002-2030-3741 (S. Xie)



these sentences may cause mental harm to individuals or certain groups. We participate in two subtasks of the German task. Subtask composition: the subtask A is a binary classification problem, which is used to determine whether the document contains hate speech, offensive content, or blasphemous words. Subtask B is a multi-category classification problem, which is used to further classify whether a document or post contains hate speech, offensive content, or blasphemous words against individuals or groups.

The rest of the paper is organized as follows: In Section 2, we introduce the related work of hate speech and offensive language detection. Section 3 gives a description of our proposed model and a summary of the dataset. Section 4 introduces the experiment and results, and section 5 gives the conclusion.

2. Related Work

Some works have been done on the classification of hate speech and offensive language. In the following, we will briefly introduce and discuss the work of these researchers. On social networks, people can interact without face-to-face, and more importantly, people often have different backgrounds and perspectives. In the anonymous environment, some users use hate speech to cause controversy to gain a sense of security. So you'll find that anonymous users on some sites are making hate statements. Nascimento et al. [3] classified Brazilian Portuguese texts to detect hate speech. In social media, people are attacked by hate speech and offensive language for various reasons, such as gender [4, 5], different nationalities [6]. Secondly, the targets of hate speech attacks are also very wide, such as Muslims [7, 8], immigrants [9], and Jews [10, 11]. Djuric et al. [12] detected hate speech by using a Logistic Regression classifier. Kamble and Joshi [13], Santosh and Aravind [14], and Mathur et al. [15] studied the classification of hate speech in Hindi and English. Nobata et al. [16] detected insulting language by using a regression model. With the deepening of research, there are more and more methods to classify hate speech and offensive language. People used logistic regression [17], hybrid volatile neural networks [18], naive Bayes [19], and other methods to classify. The biggest difficulty in classifying hate speech on social media was that it was difficult to separate hate speech from other types of aggressive language. The lexical detection methods regard the messages containing specific terms as hate speech, which results in the accuracy of the method can not meet people's requirements. Davidson et al. [20] used crowdsourcing to divide the data sets containing hate speech collected from tweets into three categories: only offensive language, containing hate speech, and not including hate speech and offensive language.

3. Methodology and Corpus

In this section, we first analyze the data used and describe the distribution of the data. Then we describe the model we used.

Table 1
Label distribution of German subtask A and German subtask B

	Label	Train set	Test set
subtask A	NOT	1700	134
	HOF	673	392
subtask B	NONE	1700	378
	HATE	146	24
	OFFN	140	36
	PRFN	146	88

3.1. Data description

We take part in the German language task. The German-language data set is collected from Twitter. Subtask A and subtask B use the same dataset. For subtask A, the dataset contains two labels: Hate and Offensive (HOF) and Non-Hate and offensive (NOT). For subtask B, the dataset contains four labels: (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 and (NOT) Not: Non-Hate and offensive. For details about the task, we refer the reader to the shared task publication [21]. Because of the relevance of label classification, it makes the classification task more difficult. Table 1 shows the detailed statistics in the dataset. As can be seen from table 1, the given dataset is imbalanced.

3.2. Model Description

BERT[22] is a bidirectional encoder representation from Transformers. It is a new language model developed and released by Google at the end of 2018. BERT model plays an important role in many natural language processing tasks, such as question answering, named entity recognition, natural language reasoning, text classification, and so on. We use the BERT-Ger(bert-base-german-cased)[1] as our pre-trained model. The BERT-Ger trained 810k steps with a batch size of 1024 for sequence length 128 and 30k steps with sequence length 512. In training, it takes about 9 days. As training data, BERT-Ger uses the latest German Wikipedia dump (6GB of raw text files), the OpenLegalData dump (2.4 GB), and news articles (3.6 GB). In the classification task, the last layer hidden state of the first token of the sequence (CLS token) is processed by a linear layer and a Tanh activation function to get the output of BERT-Ger (pooler output). However, the pooler output's summary of input semantic content is often inadequate. To let the model gain more features of semantic content, we try to solve this problem with the model architecture in Figure 1. First of all, the sequence of hidden states at the output of the last layer of the BERT-Ger is given to us, also known as the *last_hidden_state*[2]. Then, we input the *last_hidden_state* into Bi-LSTM and Bi-GRU to get the output of Bi-GRU and the hidden state of Bi-GRU(*h_n*). Thirdly, we get the *H_AVG* by average-pooling and the *H_max* by max-pooling after getting the output of Bi-GRU. Finally, we concatenate *H_max*, *H_AVG*, *h_n* and *P_O* into the classifier.

[1]https://huggingface.co/bert-base-german-cased
[2]https://huggingface.co/transformers/model doc/bert.htmlbertmodel

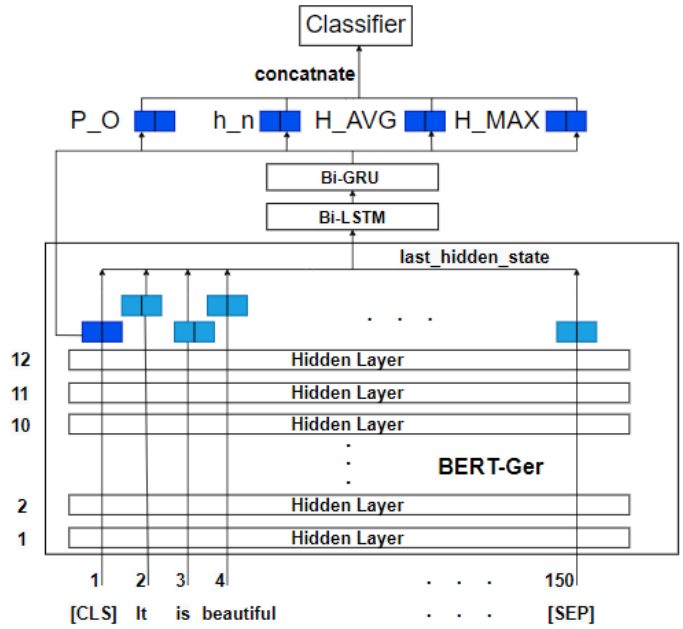


Figure 1: Model description. (*P_O*: the pooler output. *last_hidden_state*: the sequence of hidden-states at the output of the last layer of the BERT-Ger model. *h_n*: the second output returned by Bi-GRU. *H_AVG*: the average-pooling of Bi-GRU output. *H_MAX*: the max-pooling of Bi-GRU output.)

Table 2
The hyper-parameters of each subtask

	subtask A	subtask B
Hyperparameters	dropout=0.5 learning rate=1e-5 epoch=10 per gpu train batch size=4 gradient accumulation steps=4	dropout=0.5 learning rate=1e-5 epoch=10 per gpu train batch size=4 gradient accumulation steps=4

4. Experiment and results

In this section, we mainly introduce the steps of our experiment and the results of the competition.

4.1. Experiment

First of all, we get the new validation set and the new training set by using the stratified 5-fold cross-validation. The data set used is the training set provided by the competition organizers, with a total of 2373 pieces of data. This method of stratified sampling ensures that the sample proportion in each dataset remains the same. Then, in each fold of the data set, we choose the model with the highest F1 score in the validation set to predict the test set. And we get the prediction results of the model by averaging the probability of the five prediction results.

Table 3
The results of each subtask in official and private test sets

	f1-score in official test set	f1-score in private test set
subtask A	0.7961	0.5225/the best (0.5235)
subtask B	0.5409	0.2579/the best (0.2943)

Thirdly, we input data into one to four models for training with the training set and predict one to four results with the test set. Finally, we get the final result by combining the four results by hard voting.

In the experiment, we use the triangular learning rate, and the parameter learning rate is set to 1e-5. The learning rate is gradually increased through warm-up, and the linear learning rate is gradually reduced through linear learn rete decay. This experimental setup significantly improves the training effect. To save GPU memory, the gradient accumulation steps are set to 4 and the batch size parameter of GPU in fine-tuning is set to 4. The hyperparameters for each German language subtask are shown in Table 2.

4.2. Results

The official ranking is based on the test scores of private test sets. We also get the F1 Macro average according to the test sets given by the official. The specific scores are given in Table 3. After the fine-tuning of the BERT-Ger model, the ability of our model to obtain semantic information characteristics has been improved. This enables us to get a good result in the German language task. Our method ranks 2nd (2/25) in German subtask A and the F1 Macro average score is 0.5225. In German subtask B, our method ranks 10th (10/19) and the F1 Macro average score is 0.2579.

5. Conclusion

This paper introduces the model and final results of the Simon team in Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC) 2020 Challenge. The model uses BERT-Ger, Bi-GRU, Bi-LSTM, and so on, which improves the ability of the model to obtain semantic information features. The identification of hate speech and offensive content in social media has a positive effect on the development of society. In the future, we will participate in more such tasks and contribute to the identification of multilingual hate speech.

Acknowledgments

First of all, we would like to thank the HASOC sharing task organizers for giving us a happy experience and congratulations on the success of the sharing task. Secondly, I would like to thank BY for his valuable suggestions.

References

[1] Puro, Steven, Encyclopedia of the american constitution (book review)., Library Journal (2000).

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

[3] G. Nascimento, F. Carvalho, A. M. D. Cunha, C. R. Viana, G. P. Guedes, Hate speech detection using brazilian imageboards, in: the 25th Brazillian Symposium, 2019.

[4] Reddy, Vasu, Perverts and sodomites: homophobia as hate speech in africa, Southern African Linguistics Applied Language Studies 20 (2002) 163–175.

[5] C. Gatehouse, M. Wood, J. Briggs, J. Pickles, S. Lawson, Troubling vulnerability: Designing with lgbt young people's ambivalence towards hate crime reporting, in: Conference on Human Factors in Computing Systems, 2017. doi:10.1145/3173574.3173683.

[6] K. Erjavec, M. P. Kovali, "you don't understand, this is a new war!" analysis of hate speech in news web sites' comments, Mass Communication Society 15 (2012) 899–920.

[7] I. Awan, Islamophobia on social media: A qualitative analysis of the facebook's walls of hate, International Journal of Cyber Criminology 10 (2016).

[8] B. Vidgen, T. Yasseri, Detecting weak and strong islamophobic hate speech on social media, Journal of E-Government 17 (2020) 66–78.

[9] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, M. Wojatzki, Measuring the reliability of hate speech annotations: The case of the european refugee crisis (2017).

[10] M. Bilewicz, M. Winiewski, M. Kofta, A. Wójcik, Harmful ideas, the structure and consequences of anti-semitic beliefs in poland, Political Psychology 34 (2013) 821–839.

[11] J. Finkelstein, S. Zannettou, B. Bradlyn, J. Blackburn, A quantitative approach to understanding online antisemitism (2018).

[12] N. Djuric, Z. Jing, R. Morris, M. Grbovic, N. Bhamidipati, Hate speech detection with comment embeddings, in: the 24th International Conference, 2015.

[13] S. Kamble, A. Joshi, Hate speech detection from code-mixed hindi-english tweets using deep learning models, arXiv preprint arXiv:1811.05145 (2018).

[14] T. Y. S. S. Santosh, K. V. S. Aravind, Hate speech detection in hindi-english code-mixed social media text, in: the ACM India Joint International Conference, 2019.

[15] P. Mathur, R. Shah, R. Sawhney, D. Mahata, Detecting offensive tweets in hindi-english code-switched language, in: Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media, 2018, pp. 18–26.

[16] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, Y. Chang, Abusive language detection in online user content, in: the 25th International Conference, 2016.

[17] Z. Waseem, D. Hovy, Hateful symbols or hateful people? predictive features for hate speech detection on twitter, in: Naacl Student Research Workshop, 2014.

[18] J. H. Park, P. Fung, One-step and two-step classification for abusive language detection on twitter, arXiv preprint arXiv:1706.01206 (2017).

[19] H. M. Saleem, K. P. Dillon, S. Benesch, D. Ruths, A web of hate: Tackling hateful speech in online social spaces, arXiv preprint arXiv:1709.10159 (2017).

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

[21] 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.

[22] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://www.aclweb.org/anthology/N19-1423. doi:10.18653/v1/N19-1423.