ZYJ at HASOC 2020: ALBERT-Based Model for Hate Speech and Offensive Content Identification

Yingjia Zhao[a], Xin Tao[b]

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

Abstract

Online social media platforms provide convenience for people to communicate, but the harm caused by online hate speech and offensive language accompanying them is also significant. At the same time, it is a challenge to identify indirect insults such as metaphor and irony, so it is necessary to understand the semantic information of the text in depth. This paper describes the approach our team is using at HASOC2020: Hate Speech and Offensive Content Identification in Indo-European Languages. In Subtask A and Sub-task B for English, we fine-tune ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, and add a customized network structure that enables the model to take advantage of the semantic information extracted by ALBERT to complete the classification task, and use StratifiedKFold to ensemble. We achieve Marco F1 of 0.4994 and 0.2412 in Subtask A and Subtask B for English language, ranked 15th and 11th.

Keywords

Hate speech, Offensive language, ALBERT

1. Introduction

The network platform constructs a brand new living and cultural space, promotes the communication and exchange among netizens, and makes all kinds of information and speech grow exponentially in the network space. Some individuals or groups with ulterior motives take the opportunity to spread illegal information, which is not conducive to social stability or may infringe upon the legitimate rights and interests of others. Among them, online hate speech and offensive language are a kind of undesirable speech that does great harm and attracts wide attention. Online hate speech and offensive language not only make people feel uncomfortable, but also make the victim suffer from severe depression, causing irreversible psychological harm. In addition to psychological harm, such toxic online content can also lead to real hate crimes[1], therefore, automatic detection of hate speech and offensive language on social media platforms is very necessary. While direct insults and insults involving profanity are easy to identify, identifying indirect insults, for example, often involving metaphor and irony, is sometimes a challenge to human annotators, therefore, it is also a challenge for the most advanced systems[2].

HASOC2020[3] similar to HASOC2019, that is proposed for identifying hate speech and offensive content in Indo-European languages[4]. In this competition, we take part in Sub-task

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

zyj1309700118@gmail.com (Y. Zhao); taoxinwy@126.com (.X. Tao)

0000-0002-7010-8522 (Y. Zhao); 0000-0002-6883-8403 (.X. Tao)

A: Identifying Hate, offensive and profane content for English language and Sub-task B: Discrimination between Hate, profane and offensive posts for English language. We use a model based on ALBERT: A Lite BERT for Self-supervised Learning of Language Representations [5], this model combines the ALBERT model with a specific network structure to further process the eigenvectors of ALBERT output. During the training, we use the training data provided by HASOC2020 as the training data set to train the model of this task. Finally, we use the StratifiedKFold fold method to ensemble. The rest of the paper is as follows: In the second part, we cover some related work, such as the ALBERT model and some of the methods previously used to identify hate speech and offensive language. In the third part, we describe our approach, including model building and setup. In the fourth part, the results are listed and analyzed.

2. Related Work

As online social media platforms face great challenges in identifying hate speech and offensive language, the NLP community has done a lot of work on identifying hate speech, offensive language, cyberbullying, abusive content, and organized some shared tasks on these topics, such as HASOC, HatEval[6] and OffensEval[7]. To quickly and accurately identify hate speech and offensive language, industry and academia have tried a number of different architectures and approaches, some used machine learning methods combined with NLP, while others used Deep Learning (DL) [8] methods of multi-layer Neural Networks (NN) [9] stacked, and various pre-training models based on Transformers [10] were also widely used.

In HASOC 2019, IRLab@IITBHU[11] teams applied two traditional machine learning approaches: Support Vector Machine, XGBoost with a frequency-based feature for hate speech and offensive content identification, XGBoost achieved better results than SVM. YNU WB [12] teams established an ordered neuron LSTM(ON-LSTM) model with attention mechanism by deep learning method, which achieved the best results in Subtask A for English. The BRUMS[13] teams used the fine-tuned BERT[14] with simple text classifier to achieve the result second only to the YNU WB team.

ALBERT: ALBERT is a simplified model designed by Google on the basis of BERT, mainly to solve the problem of BERT with too large parameters and too slow training. ALBERT overcame the obstacles to pre-training model expansion through two parameter reduction techniques: Factorized embedding parameterization, the large word embedded matrix is decomposed into two small matrices, so as to separate the size relation between the hidden layer and the dictionary, the two are no longer directly related, so that the node number expansion of the hidden layer is no longer limited. Cross-layer parameter sharing, this prevents the number of arguments from increasing with the depth of the network. Both techniques significantly reduce the number of arguments without significantly affecting their performance. The ALBERT configuration is similar to the BERT-Large level, but with an 18-fold reduction in the number of arguments and a 1.7-fold increase in training speed. At the same time, parameter reduction also plays a regularization role, which greatly enhances the generalization ability of the model.

3. Methodology and Data

3.1. Data description

For this task, we use data sets provided by HASOC2020, mainly from Twitter. This task is divided into two subtasks. Sub-task A: Identifying Hate, offensive and profane content. Sub-task A focus on Hate speech and Offensive language identification offered for English, German, Hindi. Sub-task A is coarse-grained binary classification in which participating system are required to classify tweets into two class, namely: Non- Hate and offensive (NOT): This post does not contain any Hate speech, profane, offensive content , and Hate and Offensive (HOF): This post contains Hate, offensive, and profane content. Sub-task B: Discrimination between Hate, profane and offensive posts. This sub-task is a fine-grained classification offered for English, German, Hindi. Hate-speech and offensive posts from the sub-task A are further classified into three categories. Hate speech (HATE), Offensive(OFFN) and Profane(PRFN), here, we need to add another category, (NONE) : posts that do not contain any of the above. In this task, we take part in the English task. For English language, the training dataset has a total of 3708 data, of which there are 1856 of HOF and 1852 of NOT, and there are 159 of HATE, 321 of OFFN, 1377 of PRFN and 1852 of NONE.

3.2. Our model

Our model structure is shown in Figure 1. First, we input the preprocessed text into the model through the input layer, and then vector representation is carried out. Vector representation is divided into three parts: word vector representation, text vector representation and position vector representation. ALBERT model converts each word in the text into the vector by querying the word vector table, namely, word vector representation; the value of this vector is automatically learned in the model training process, which is used to describe the global semantic information of text and integrate with the semantic information of words, namely, text vector representation; because the semantic information carried by words appearing at different positions in the text is different, the ALBERT model attaches a different vector to the words at different positions to make a distinction, namely, position vector representation. Then ALBERT model takes the addition of word vector, text vector and position vector as input, which is further processed by the Transformer Encoder module, we then input the output of the last hidden layer of ALBERT into BiLSTM, and concatenate the output of BiLSTM with the output of the last hidden layer of ALBERT, and then, through the Relu function, map the splited vector to the lower dimension in a nonlinear way. After that, max-pooling will be used to take the maximum value of each position in the vector on all timing sequence to obtain the feature vector. Finally, the feature vector will concatenate with the original ALBERT output and input into the classifier.

3.3. StratifiedKFold ensemble

In this experiment, we use the StratifiedKFold ensemble method to train different data sets during each training process, and extract different features during the process of extracting features from the model, so as to enhance the generalization ability of the model and achieve

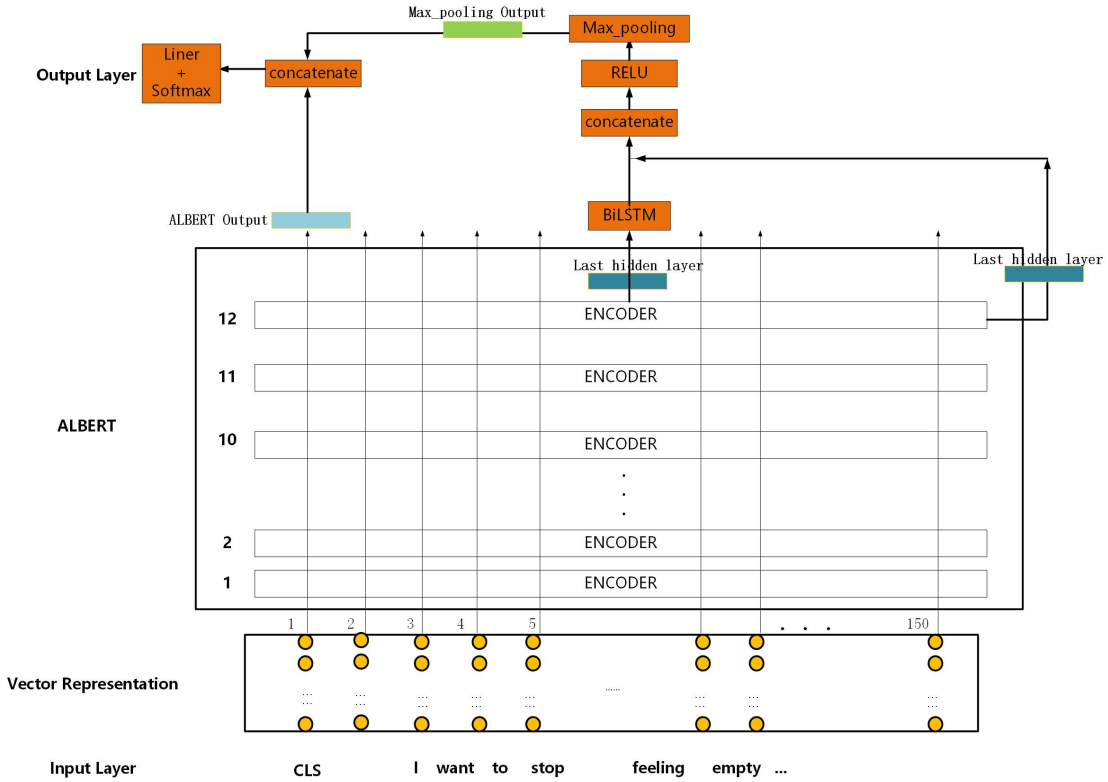

Figure 1: Schematic overview of the architecture of our model

the purpose of enhancing the performance of the model. The idea of StratifiedKFold is *K*-fold cross-segmentation. Each type of data in the initial training set is divided into *K* sub-samples, a single subsample is retained as the data of the verification model, and the other *K-1* samples are used for training, that is, to ensure the hierarchical sampling, and the proportion of all kinds of samples in the training set and test set is the same as that in the original data set, as shown in Figure 2.

4. Experiment and results

4.1. Data preprocessing

In order to eliminate the interference of irrelevant information in the tweet, so that the model can better extract text features and train the model more efficiently, we preprocessed the original training data provided by the official with NLTK tool. The main processing steps are as follows:

- Remove number. Numbers generally have no meaning in text analysis, so they need to be removed before further analysis.

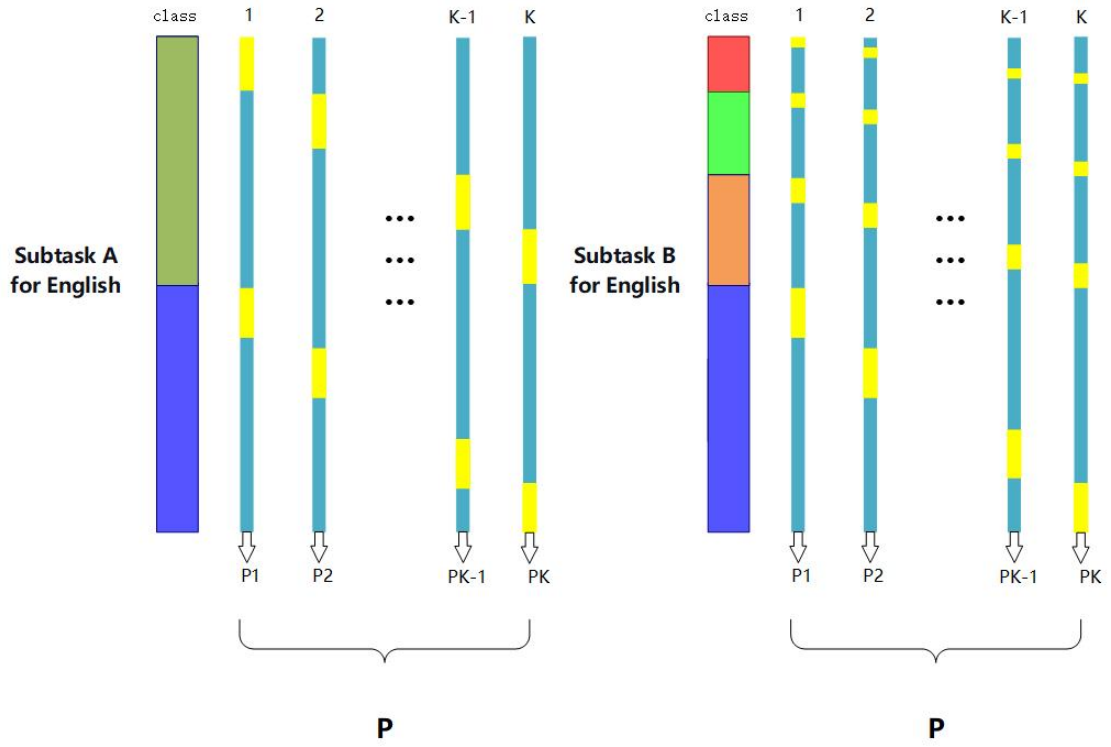

Figure 2: The StratifiedKFold ensemble approach

- Steming. The process of reducing the derived form of a word to its stem, which will greatly shorten the word list and improve the efficiency.
- Remove link address, whitespace, etc.(e.g https://t.co/5u8Di1waFC.)
- Remove special characters. Characters that occur frequently but are meaningless for training need to be removed. (Remove the 'RT' from the sentence.)
- Converts all characters to lowercase.

4.2. Experiment setting

The pre-training model we use in this experiment is albert_base_v2 from the ALBERT series model. In the process of fine-tuning, the maximum sequence length is set to 150, the learning rate is set to 2e-5, the gradient accumulation steps and batch size are both set to 4. By using 5-fold crossvalidation on the training data, we set the epoch to 10 for training, and the best weight is saved during the training.

Table 1
The result we submitted in **Subtask A** for English language

Rank	Team	F1 Macro average
1	IIIT_DWD	0.5152
2	CONCORDIA_CIT_TEAM	0.5078
3	AI_ML_NIT_Patna	0.5078
4	Oreo	0.5067
5	MUM	0.5046
...		
15	**ZYJ**	**0.4994**

Table 2
The result we submitted in **Subtask B** for English language

Rank	Team	F1 Macro average
1	chrestotes	0.2652
2	hub	0.2649
3	zeus	0.2619
4	Oreo	0.2529
5	Fazlourrahman Balouchzahi	0.2517
...		
11	**ZYJ**	**0.2412**

4.3. Results

The organizers evaluate the classification system by calculating the Marco F1 score. According to the official results, our team' s Marco F1 score is 0.4994, ranked 13th place in Subtask A for English language, as shown in Table 1. And in Subtask B for English language, our team' s Marco F1 score is 0.2412, ranked 10th place, as shown in Table 2.

5. Conclusion

In this paper, we describe our work in HASOC2020: Identifying hate speech and offensive language. Our model uses ALBERT pre-training model to extract the semantic information features of text, and further processes the output features by using customized network structure. Finally, StratifiedKFold ensemble is used to improve the generalization ability of the model, and there are fewer model parameters, making the model lighter and easier to train. Our model achieves satisfying performance. In further research work, we will try to fine-tune ALBERT's more hidden layers to make the model more suitable for specific training tasks, thus further improving the model performance.

Acknowledgments

We would like to thank the organizers for organizing this shared task and the teachers for their help. Finally, we would like to thank the school for its support to my research and the future reviewers for their patient work.

References

[1] M. J. Matsuda, Public response to racist speech: Considering the victim's story, Michigan Law Review 87 (1989) 2320–2381.

[2] S. Malmasi, M. Zampieri, Challenges in discriminating profanity from hate speech, Journal of Experimental & Theoretical Artificial Intelligence 30 (2018) 187–202.

[3] 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.

[4] 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.

[5] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, arXiv preprint arXiv:1909.11942 (2019).

[6] V. Basile, C. Bosco, E. Fersini, N. Debora, V. Patti, F. M. R. Pardo, P. Rosso, M. Sanguinetti, et al., Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter, in: 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, 2019, pp. 54–63.

[7] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval), arXiv preprint arXiv:1903.08983 (2019).

[8] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (2015) 436–444.

[9] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, J. Cernocky, Rnnlm-recurrent neural network language modeling toolkit, in: Proc. of the 2011 ASRU Workshop, 2011, pp. 196–201.

[10] 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.

[11] A. Saroj, R. K. Mundotiya, S. Pal, Irlab@ iitbhu at hasoc 2019: Traditional machine learning for hate speech and offensive content identification., in: FIRE (Working Notes), 2019, pp. 308–314.

[12] B. Wang, Y. Ding, S. Liu, X. Zhou, Ynu_wb at hasoc 2019: Ordered neurons lstm with attention for identifying hate speech and offensive language., in: FIRE (Working Notes), 2019, pp. 191–198.

[13] T. Ranasinghe, M. Zampieri, H. Hettiarachchi, Brums at hasoc 2019: Deep learning models

for multilingual hate speech and offensive language identification., in: FIRE (Working Notes), 2019, pp. 199–207.
[14] 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).