

# ALBERT for Hate Speech and Offensive Content Identification

Jun Zeng, Li Xu and Hao Wu\*

*School of Information Science and Engineering, Yunnan University, Kunming, P.R. China*

## Abstract

This paper describes our system in Subtask 1A of HASOC 2021, and our team name is JZ2021. Subtask 1A focuses on hate speech and offensive language recognition for English and Hindi. Now, the detection of hate speech and offensive content on the Internet has received widespread attention. These comments have caused a lot of trouble to people, and the identification of the comments are very meaningful. With the development of deep learning, many pre-trained deep neural network models are used for text classification tasks. However, some pre-trained models contain a large number of parameters, although they perform well. In HASOC 2021 task, we use a model called ALBERT. It improves the BERT model and effectively reduces the number of parameters of the model. We chose ALBERT-large, which gets great results in the task. Our system achieves the Macro F1 score of 83.75%.

## Keywords

Hate Speech, Offensive Content, HASOC 2021, ALBERT, BERT

## 1. Introduction

In recent years, with the rapid development of Internet technology, the number of users is rapidly increasing, and various social platforms have also emerged. Netizens can freely express their opinions on the platforms. These platforms mainly have anonymous functions, so many people will give vent to their dissatisfaction of life. As a result, many hate speeches or offensive contents have been generated on the Internet. This kind of problem needs to be taken seriously, because it will not only cause distress to people, but even cause some people to suffer from mental illness or suicide. However, there are a huge amount of comments generated on the Internet every day, and it is very unrealistic to recognize these comments by manual methods. It is particularly important to use artificial intelligence methods to replace manual methods.

The identification of hate speech and offensive content faces some challenges. First of all, posts on social media include multiple languages, and each person's writing style is different. At the same time, the irregular writing and the emergence of some new Internet expression will also bring some difficulties to the detection task. Secondly, some comments do not directly contain insulting words, but are implicit or ironic attacks. This is also a difficult point. In addition, people do not have a very clear standard for the definition of hate speech. The performance of the model highly depends on the training data set, which is related to the person who mark

---


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

✉ zengjun@mail.ynu.edu.cn (J. Zeng); x619496775@gmail.com (L. Xu); haowu@ynu.edu.cn (H. Wu\*)

🆔 0000-0002-9269-3443 (J. Zeng); 0000-0001-5130-1645 (L. Xu); 0000-0002-3696-9281 (H. Wu\*)



© 2021 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)

the data to a certain extent.

In response to the above issues, the NLP community has released a series of tasks on hate speech identification. HASOC 2021 shared task is one of them [1] [2]. It is dedicated to the identification of hate speech and offensive content in English and Indo-Aryan languages. In this task, we used the pre-training model ALBERT [3], which has made some improvements on the basis of the BERT model [4], it greatly reduces the number of parameters and improves the speed of training.

The structure of this article is as follows: Section 2 introduces related research on hate speech and offensive speech recognition. Section 3 explains the model used in this task. Section 4 shows the experimental procedure. Section 5 describes the results of the experiment and the analysis of the results. Section 6 summarizes this work.

## 2. Related Works

Identifying hate speech is a text classification [5] task, and text classification has attracted a lot of attention in the field of natural language processing. In recent years, deep learning technology has developed rapidly and has been widely used in many fields. In text classification tasks, deep learning models based on Convolutional Neural Networks (CNN) [6], Recurrent Neural Networks (RNN) and Attention Mechanisms have made good progress.

CNN were initially successful in the image field [7], and they were used for text classification [8] later. TextCNN [9] is a deep learning model based on CNN, which has achieved excellent results in classification tasks. CNN cannot process sequential input data, but in natural language processing tasks, most of the input data is sequential data. In order to solve this demand, Recurrent Neural Network (RNN) has also developed rapidly. Long Short-Term Memory networks (LSTM) [10] and GRU are two classic RNN-based models, but they can only handle fixed-length sequences. Sutskever et al. proposed the Sequence to Sequence (Seq2Seq) [11] model, which can handle variable-length sequences and is used for machine translation. Later, the Attention Mechanism was proposed, and many models began to adopt the Attention Mechanism. Google proposed the Transformer model [12], which only uses the Attention Mechanism. The proposal of Transformer has brought far-reaching influence to the field of natural language processing. In order to get better results, the current model parameters are increasing considerably, and training become slower. Pre-trained models can alleviate this problem. Model can adapt some tasks by fine-tuning. GPT and BERT are two pre-training models, both of which are based on the Transformer structure. After the release of BERT, many new results have appeared in the NLP task. The BERT model has a huge amount of parameters, which brings some difficulties to training. ALBERT model [3] alleviates these problems, and its performance is also very great.

## 3. Methods

### 3.1. BERT

BERT [13]: Bidirectional Encoder Representation from Transformers, just like its name, this is a bidirectional model based on Transformers. BERT uses a large amount of text data to construct

**Table 1**

Parameters of two versions of BERT model

Model	L	H	A	TotalParameters
BERT-base	12	768	12	110M
BERT-large	24	1024	16	340M

two pre-training tasks, namely Masked Language Model (MLM) and Next Sentence Prediction (NSP). MLM alleviates the constraint of single direction. It randomly masks some tokens in the input, and then predicts the masked words based on the context. This is different from the traditional left-to-right training model, because it allows us to generate deep bidirectional language representations. When using MLM, we don't always use the actual [mask] token to replace the masked word. The training data generator will randomly select 15% of the token positions for prediction. The selected token will perform the following operations: (1) Replacing the token with the [mask] token 80% of the time; (2) Replacing the token randomly with a token 10% of the time; (3) Remaining unchanged 10% of the time. NSP is used to train a model so that the model can understand the relationship between sentences, because it is very important for many downstream tasks to understand the relationship between two sentences. Specifically, we select sentence  $A$  and sentence  $B$  in the corpus to form a training example.  $B$  is the next sentence of  $A$  at the time of 50%, and 50% of the time is not. The main structure of BERT is stacked by the encoder of the Transformer. It takes a series of words as input, and applies the Self-attention Mechanism to each layer, and then passes the result to the next encoder through the feedforward neural network. BERT is divided into two versions: BERT-base and BERT-large, as shown in Table 1.  $L$  represents the number of layers of the Transformer,  $H$  represents the dimension of the output,  $A$  represents the number of Multi-head Attention, and TotalParameters represents the size of the model parameters. Because of the Self-attention Mechanism in Transformer, BERT can be well used for many downstream tasks by fine-tuning.

### 3.2. ALBERT

Recently, the trend in the NLP field is to use large-scale models to obtain better performance. However, stacking model parameters brainlessly may not bring better results. Although BERT is powerful, its parameters are very large, which puts forward higher requirements on hardware conditions, and training also consumes more time. The emergence of A Lite BERT (ALBERT) alleviates this problem, and its parameters are significantly less than the traditional BERT model.

ALBERT mainly uses two methods to reduce model parameters. The first way is factorized embedding parameterization. In BERT model, the WordPiece [13] embedding size  $E$  and the hidden layer size  $H$  are equal. However, this approach is not necessary. In reality, NLP requires a large vocabulary  $V$ , and the size of the embedding matrix is  $V \times E$ . If  $E$  is always equal to  $H$ , increasing  $H$  will cause the embedding matrix to increase. As a result, the parameters of the model may increase dramatically. In ALBERT, factorization is used to decompose the embedded parameters into two smaller matrices. First, we project the one-hot vector into a low-dimensional space of size  $E$  and then project it into the hidden space. Through factorization, the embedding parameter changes from  $O(V \times H)$  to  $O(V \times E + E \times H)$ . When  $H$  is much larger

**Table 2**  
Basic hyperparameters of BERT model and ALBERT model

Model	Parameters	L	H	Parameter-sharing
BERT-base	110M	12	768	False
BERT-large	340M	24	1024	False
ALBERT-base	12M	12	768	True
ALBERT-large	18M	24	1024	True
ALBERT-xlarge	60M	24	2048	True
ALBERT-xxlarge	235M	12	4096	True

than  $E$ , the parameters of ALBERT are reduced a lot compared to BERT. Another method is cross-layer parameter sharing. In order to further improve parameter efficiency, ALBERT uses a cross-layer parameter sharing method. There are many ways to share parameters. But the default way is to share all parameters across layers in ALBERT.

BERT hope the model will learn to understand the relationship between two sentences by using the NSP loss, so that the model can adapt to NLP tasks like QA. However, research has found that the effect of NSP is not good, and this method is unreliable. ALBERT proposed sentence-order prediction (SOP) loss, which emphasizes the coherence between sentences. The SOP loss takes two consecutive segments from the same document as a positive example, and a negative example swaps the positions of the two consecutive segments. NSP tends to learn simpler topic prediction signals, so it cannot solve the SOP task, but SOP has a good performance on the NSP task.

There are 4 versions about ALBERT: ALBERT-base, ALBERT-large, ALBERT-xlarge and ALBERT-xxlarge. The basic hyperparameters of the different versions of the BERT model and the ALBERT model are shown in Table 2. Obviously, when the layers and hidden layer size of ALBERT are similar to the BERT’s, the number of parameters of ALBERT is much smaller than that of BERT.

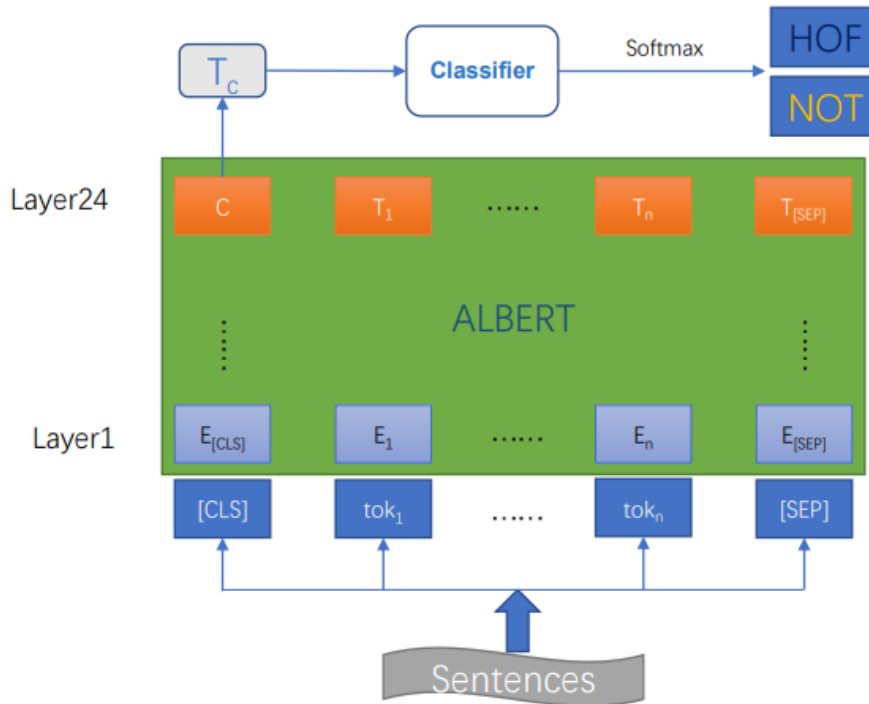
In this shared task, we chose the ALBERT model to identify hate speech and offensive content. More specifically, we used ALBERT-large, its parameter size is 18M, there are a total of 24 Transformer layers,  $Hidden = 1024$ . The reason for introducing the BERT model is that ALBERT is improved based on BERT, and we need to compare the parameters of BERT and ALBERT. The smaller amount of parameters is the reason why we choose ALBERT.

## 4. Experiment

This section will introduce task description, the data set we used, and other experimental details.

### 4.1. Task Description

We participated in Subtask 1A [1]. Subtask 1A focuses on hate speech and offensive language recognition for English and Hindi. It is a coarse-grained binary classification, in which the participating system is required to classify tweets into two classes, namely: Hate and Offensive (HOF) and Non-Hate and offensive (NOT). The definition of NOT is that this post does not



**Figure 1:** The architecture of our model

contain any hate speech, profanity, or offensive content. The definition of HOF is that this post contains hate, offensive, and profane content.

#### 4.2. Dataset

The data of this experiment is provided by HASOC 2021 [2], and the data sets are from Twitter. We used the English data set for related tasks. There are a total of 3790 English training data instances, of which the number of labels with HOF is 2417, and the number of NOT is 1325. The number of instances in the test data set is 1268.

#### 4.3. Experimental Steps

First of all, we have to preprocess the data. This mainly includes: (1) converting Emojis into corresponding phrases; (2) changing all text to lowercase form; (3) turning numbers into string form; (4) removing @ symbol. We divide 80% of the English training data set into training data, and use the rest as validation data. We use ALBERT for embedding.  $T_C$  ([CLS] vectors) contains the semantic information of the entire sentence, it is transferred to the classifier (fully connected layers), and then activated using the softmax function. The loss function is `sparse_categorical_crossentropy`, and as optimizer we use Adam. The architecture of our model is shown in Figure 1.

**Table 3**

The result of the English dataset under the test set

Model	Acc(%)	Macro F1(%)
SVM	66.42	65.03
LR	63.90	63.01
BERT-base	83.75	83.26
ALBERT-large	<b>83.92</b>	<b>83.75</b>

## 5. Results

We also use other methods as baseline models. They are SVM [14], LR (Logistic Regression) [15] and BERT-base. SVM is a binary classifier model that maps the feature vector of the example into some points in the space. The purpose of SVM is to draw a line to correctly distinguish these points. LR is also a binary classifier method. We use Macro F1 to evaluate the performance of the model, and results are shown in Table 3.

It can be seen that the performance of ALBERT-large is the best (Macro F1 of 83.75%), and we use the model for submitting runs to the shared task. BERT-base also performed well (Macro F1 of 83.26%, which is 0.49% worse than ALBERT-large), ranking second among the models. The performance of the other two traditional machine learning methods is vastly worse. Deep learning methods can automatically extract feature with neural networks. Although the computational cost is higher, it also gets more useful information, and its performance is better than traditional machine learning methods.

## 6. Conclusion

Now, the performance of deep learning models is constantly improving, but the number of parameters is also increasing considerably. ALBERT effectively reduces the number of model parameters through factorization of the embedding layer and cross-layer parameter sharing, so that ordinary users can also run it. Compared to traditional machine learning methods, ALBERT performs better. ALBERT has a performance that is not inferior to BERT, and the number of parameters is much smaller. In this task, unfortunately, due to our insufficient hardware conditions, we cannot try the larger versions of ALBERT-xlarge and ALBERT-xxlarge.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (61962061, 61562090, U1802271), partially supported by the Yunnan Provincial Foundation for Leaders of Disciplines in Science and Technology(202005AC160005), Top Young Talents of "Ten Thousand Plan" in Yunnan Province(YNWR-QNBJ-2019-188), the Program for Excellent Young Talents of Yunnan University.

## References

- [1] 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/>.
- [2] 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.
- [3] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020. URL: <https://openreview.net/forum?id=H1eA7AEtvS>.
- [4] M. T. R. Laskar, E. Hoque, J. X. Huang, Utilizing bidirectional encoder representations from transformers for answer selection, CoRR abs/2011.07208 (2020). URL: <https://arxiv.org/abs/2011.07208>. arXiv:2011.07208.
- [5] D. Alsaleh, S. L. Marie-Sainte, Arabic text classification using convolutional neural network and genetic algorithms, IEEE Access 9 (2021) 91670–91685. URL: <https://doi.org/10.1109/ACCESS.2021.3091376>. doi:10.1109/ACCESS.2021.3091376.
- [6] H. S. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, N. A. Ajlan, Classification of remote sensing images using efficientnet-b3 CNN model with attention, IEEE Access 9 (2021) 14078–14094. URL: <https://doi.org/10.1109/ACCESS.2021.3051085>. doi:10.1109/ACCESS.2021.3051085.
- [7] G. Dong, Y. Ma, A. Basu, Feature-guided CNN for denoising images from portable ultrasound devices, IEEE Access 9 (2021) 28272–28281. URL: <https://doi.org/10.1109/ACCESS.2021.3059003>. doi:10.1109/ACCESS.2021.3059003.
- [8] C. Peng, T. Bao, An analysis method for interpretability of CNN text classification model, Future Internet 12 (2020) 228. URL: <https://doi.org/10.3390/fi12120228>. doi:10.3390/fi12120228.
- [9] I. Alshubaily, Textcnn with attention for text classification, CoRR abs/2108.01921 (2021). URL: <https://arxiv.org/abs/2108.01921>. arXiv:2108.01921.
- [10] C. Acartürk, M. Sirlanci, P. G. Balikcioglu, D. Demirci, N. Sahin, O. A. Kucuk, Malicious code detection: Run trace output analysis by LSTM, IEEE Access 9 (2021) 9625–9635. URL: <https://doi.org/10.1109/ACCESS.2021.3049200>. doi:10.1109/ACCESS.2021.3049200.
- [11] S. Huang, X. Zhou, S. P. Chin, Application of seq2seq models on code correction, Frontiers Artif. Intell. 4 (2021) 590215. URL: <https://doi.org/10.3389/frai.2021.590215>. doi:10.3389/frai.2021.590215.
- [12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017,



December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008. URL: <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>.

- [13] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: <https://doi.org/10.18653/v1/n19-1423>. doi:10.18653/v1/n19-1423.
- [14] S. Liang, A. Q. M. Sabri, F. Alnajjar, C. K. Loo, Autism spectrum self-stimulatory behaviors classification using explainable temporal coherency deep features and SVM classifier, IEEE Access 9 (2021) 34264–34275. URL: <https://doi.org/10.1109/ACCESS.2021.3061455>. doi:10.1109/ACCESS.2021.3061455.
- [15] W. Ksiazek, M. Gandor, P. Plawiak, Comparison of various approaches to combine logistic regression with genetic algorithms in survival prediction of hepatocellular carcinoma, Comput. Biol. Medicine 134 (2021) 104431. URL: <https://doi.org/10.1016/j.combiomed.2021.104431>. doi:10.1016/j.combiomed.2021.104431.