BNU-HKBU UIC NLP Team 2 at SemEval-2019 Task 6: Detecting Offensive Language Using BERT model

Zhenghao Wu **Hao Zheng** **Jianming Wang** **Weifeng Su** **Jefferson Fong**

{1630003054,1630003067,1630003049}@mail.uic.edu.hk
{wfsu,jeffersonfong}@uic.edu.hk

Computer Science and Technology, Division of Science and Technology
BNU-HKBU United International College
Zhuhai, Guangdong, China

Abstract

In this study we deal with the problem of identifying and categorizing offensive language in social media. Our group, BNU-HKBU UIC NLP Team2, use supervised classification along with multiple version of data generated by different ways of pre-processing the data. We then use the state-of-the-art model Bidirectional Encoder Representations from Transformers, or BERT (Devlin et al. (2018)), to capture linguistic, syntactic and semantic features. Long range dependencies between each part of a sentence can be captured by BERT's bidirectional encoder representations. Our results show 85.12% accuracy and 80.57% F1 scores in Subtask A (offensive language identification), 87.92% accuracy and 50% F1 scores in Subtask B (categorization of offense types), and 69.95% accuracy and 50.47% F1 score in Subtask C (offense target identification). Analysis of the results shows that distinguishing between targeted and untargeted offensive language is not a simple task. More work needs to be done on the unbalance data problem in Subtasks B and C. Some future work is also discussed.

1 Introduction

Social media is an essential part of human communication today. People can share their opinions in this platform with anonymity. Some people use offensive language and hate speech casually and frequently without taking any responsibility for their behavior. For this reason, SemEval 2019 (Zampieri et al. (2019b)) set up the task OffensEval: identifying and categorizing offensive language in social media. This task is divided into three subtasks: offensive language identification, automatic categorization of offensive types, and offence target identification.

Our group uses the Natural Language Processing (NLP) latest model, Bidirectional Encoder Representations from Transformers (BERT). It is a general-purpose "language understanding" model trained on a large text corpus such as Wikipedia (Devlin et al. (2018)). After fine-tuning, the model can be used for downstream NLP tasks. Because BERT is very complex and is the state-of-art model, it is prudent for us not to change its internal structure. Hence, we focus on preprocessing the data and error analysis. After much experimentation with the data, such as translating emoji into words, putting more weight on some metaphorical words, removing the hashtag and so on, we find that using the original data will give the best performance. The reason for this is perhaps if we remove some information from the sentence, some features that affect the prediction result will be lost. So we end up using the original data to train our model.

2 Related Work

Much research has been done in detecting offensive language, aggression, and hate speech in user-generated content. In recent years, researches tend to follow several approaches: use a simple model with logistic regression to perform detection, use a neural network model, or use some other methods.

For the simple model, Davidson and Warmsley (Davidson et al. (2017)) used a sentiment lexicon designed for social media to assign sentiment scores to each tweet. This is an effective way to identify potentially offensive terms. Then they use logistic regression with $L2$ regularization to detect hate speech in social network.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 551–555
Minneapolis, Minnesota, USA, June 6–7, 2019.

Neural network models use n-gram, skip-gram or some other methods to extract features from the data. These features are used to train different models. The results produced by these models will be used as the input for training the meta-classifier (e.g. Malmasi and Zampieri (2018))

For other methods, using bag-of-words is an effective way to detect hate speech, but it is difficult to distinguish hate speech from text with offensive words that are not hate speech (Kwok and Wang (2013)). For identifying the targets and intensity of hate speech, syntactic features method is a good method (Burnap and Williams (2015)).

3 Methodology and Data

Only the training data provided by the organizer (Zampieri et al. (2019a)) are used in training our model. The data contain 13,240 pieces of tweet that had been desensitized (replacing the user names and website URLs). There are three labels that are labeled with crowdsourcing for each of the three subtasks. Gold labels obtained through crowdsourcing are confirmed by three annotators. We segmented the training set by 90% for the training set, 5% for the cross-validation set, and 5% for the test set.

Because some offensive language is subtle, less ham-fisted, and sometimes cross sentence boundary, the model trained for this task must make full use of the whole sentence content in order to extract useful linguistic, syntactic and semantic features which may help to make a deeper understanding of the sentences, while at the same time less subjected by the noisiness of speech. So, we use BERT in all three subtasks. Unlike most of the other methods, BERT uses bidirectional representation to make use of the left and right context to gain a deeper understanding of a sentence by capturing long range dependencies between each part of the sentence.

The uncased base version of the pre-trained model files [1] is used during the entire training. The training data are processed in many ways to fine-tune the model. Processing methods include removing all username tags, URL tags and symbols, converting all text to lowercase, and translating emoji into text[2]. One or more of the above methods is selected to process the training data, and then use the processed data to train the model.

In Subtask A, the accuracy after the various operations is shown in the following table.

Preprocessing	Accuracy
Original Data	0.8184
Remove tag & symbols	0.8126
Emoji translation v1	0.8081
Emoji translation v2	0.7960

Table 1: Training results for Sub-task A.

After all attempts, the best performing model for Subtask A is the model trained by the original data. Therefore, the original data are also used in the training of the Subtasks B and C models.

4 Results

For Subtask A, The BERT-Base, Uncased, original training data model get macro F1 score of 0.8057 and total accuracy of 0.8512.

For Subtask B, The BERT-Base, Uncased, original training data model get macro F1 score of 0.50 and total accuracy of 0.8792.

For Subtask C, The BERT-Base, Uncased, original training data model get macro F1 score of 0.5047 and total accuracy of 0.6995.

Results table and confusion matrices for Subtasks A, B and C are shown below.

[1] BERT-Base, Uncased: https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip

[2] In the process of translating emoji characters, v1 and v2 methods were used. **v1**: Translate all emoji characters into official character name listed in the Unicode®11.0.0 Standard. **v2**: In addition to "v1" of processing of all emoji characters, the selected 97 emotional emoji characters are translated into manually determined emotional words.

System	F1 (macro)	Accuracy
All NOT baseline	0.4189	0.7209
All OFF baseline	0.2182	0.2790
BERT-Base, Uncased, original training data	**0.8057**	**0.8512**

Table 2: Results for Sub-task A.

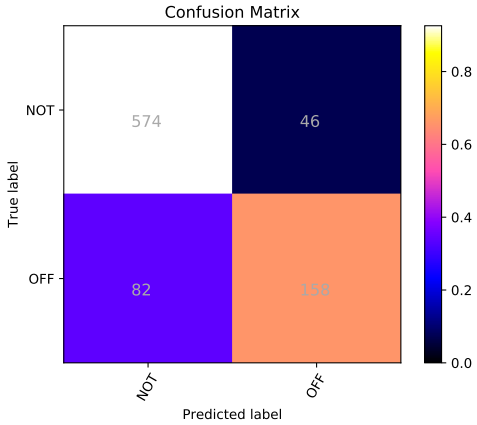

Figure 1: Sub-task A, BNU-HKBU UIC NLP Team 2 CodaLab 527070 BERT-Base, Uncased, original training data

System	F1 (macro)	Accuracy
All TIN baseline	0.4702	0.8875
All UNT baseline	0.1011	0.1125
BERT-Base, Uncased, original training data, 0.5 threshold	**0.5000**	**0.8792**
BERT-Base, Uncased, orginal training data, 0.65 threshold	0.4702	0.8875

Table 3: Results for Sub-task B.

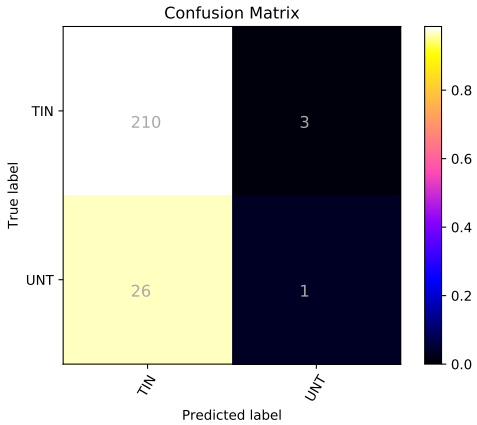

Figure 2: Sub-task B, BNU-HKBU UIC NLP Team 2 CodaLab 531958 BERT-Base, Uncased, original training data

System	F1 (macro)	Accuracy
All GRP baseline	0.1787	0.3662
All IND baseline	0.2130	0.4695
All OTH baseline	0.0941	0.1643
BERT-Base, Uncased, original training data	**0.5047**	**0.6995**

Table 4: Results for Sub-task C.

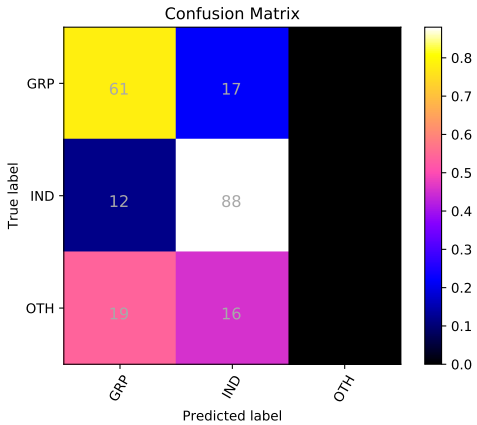

Figure 3: Sub-task C, BNU-HKBU UIC NLP Team 2 CodaLab 535873 BERT-Base, Uncased, original training data

5 Conclusions

Our model for Subtask A ranked 6 out of the 103 groups. This shows that BERT can identify offensive language. However, the results for Subtasks B and C are not as good. We will try to explain the possible reasons using error analysis.

First, some speech may be miss-labeled by annotators. In our test set, some predictions were judged as wrong, but our manual examination shows the predictions seem correct. For example, *"@USER B***hhhhh I'm jell"* and *"@USER Crazy Russian dude owns all your data"* are both labeled **NOT** (not offensive). The model, as well as our manual examination, deem these as offensive.

Second, we also notice a problem is that it is hard for our model to understanding some specific noun such as people name when our training data is not enough. For example, our model predict sentence *"Hitler will be so proud of David Hogg"* as not offensive. The word "Hitler" has a very special meaning that can makes an otherwise innocent sentence to be offensive. Our model presently can't detect this.

Another problem is emoji characters in offensive languages, which usually contains strong emotions. And may be used to express irony. So emoji characters are translated by two methods[2] to help BERT model understand the meaning of tweet posts. But the results show that both translation methods lead to a drop in accuracy. The main reason should be that some emoji characters contain different meanings in different contexts. For example, 🙂 (Slightly Smiling Face) can contain emotion of happy but also banter as well. Thus, it is difficult to understand the meaning of emoji characters in context.

Moreover, unbalanced data is a big problem. In Subtask B, few sentences are predicted as untargeted, and in Subtask C, no sentence is predicted as in the Others category. This leads to a low F1 score in these subtasks. Over-sampling in less numerous categories would not work not well in our task, and threshold moving only slightly raises the F1 score. To deal with this problem as future work, we may have to remove the labels and use unsupervised learning.

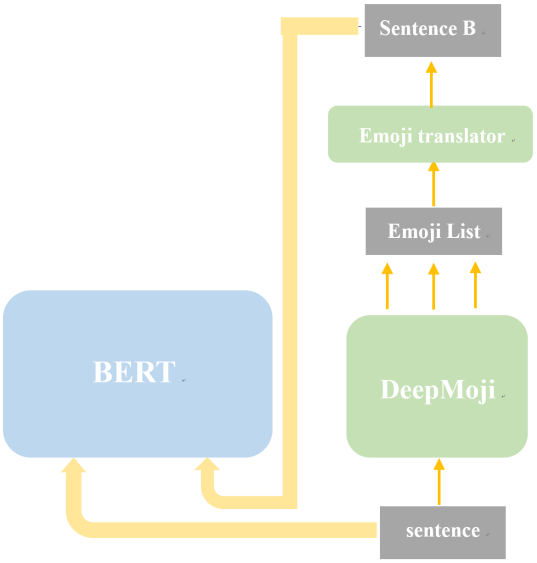

Figure 4

For future work, we notice that offensive languages often contain strong emotions such as angry, banter or taunt. This emotion and other useful contents may be improved by using DeepMoji (Felbo et al. (2017)), which translates a sentence into an emoji list to express a sentence's hidden information, such as sentiment and sarcasm. A list of emoji related to the meaning of a sentence produced by DeepMoji can be used to help BERT to better classify the sentence categories, as show in the Figure 4. The last step is to put the original sentence and the encoded new sentence as input for BERT's sentence-pair classification task.

References

Pete Burnap and Matthew L. Williams. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *ICWSM*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions

of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *AAAI*.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in discriminating profanity from hate speech. *J. Exp. Theor. Artif. Intell.*, 30:187–202.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval)*.