PGSG at SemEval-2020 Task 12:
BERT-LSTM with Tweets' pretrained model and Noisy Student training method

Bao-Tran Pham-Hong
PropertyGuru Group, Singapore
phamtrancsek12@gmail.com

Setu Chokshi
PropertyGuru Group, Singapore
setu@propertyguru.com.sg

Abstract

The paper presents a system developed for the SemEval-2020 competition Task 12 (OffensEval-2): Multilingual Offensive Language Identification in Social Media. We achieve **the second place (2nd)** in sub-task B: *Automatic categorization of offense types* and are ranked 55th with a macro F1-score of 90.59 in sub-task A: *Offensive language identification*. Our solution is using a stack of BERT and LSTM layers, training with the Noisy Student method. Since the tweets data contains a large number of noisy words and slang, we update the vocabulary of the BERT large model pre-trained by the Google AI Language team. We fine-tune the model with tweet sentences provided in the challenge.

1 Introduction

Inappropriate and offensive online content has become a significant issue due to an exponential increase in the use of the Internet by people from different cultures and educational backgrounds. Twitter is one of the most popular social media platform, where people share their own opinions among various topics. Therefore, 'tweets' requires considerable resources to study offensive behaviors.

The SemEval-2020 competition Task 12 is a multilingual challenge of 5 languages: Arabic, Danish, English, Greek, and Turkish (Zampieri et al., 2020). We participate in the English language track, sub-task A and sub-task B. In this language track, the organizers provide a data set of more than 9 million sentences of tweets along with their confidence measures produced by unsupervised learning methods (Rosenthal et al., 2020).

To handle this semi-supervised learning task, we use the Noisy Student training method (Xie et al., 2019) to train the BERT-LSTM model. Our approach is more successful in sub-task B, where the standard deviation range of the provided label's confidence is much larger, with 4.5 points better than the next system. We have publically released the code and our Tweet's pre-trained model at https://github.com/phamtrancsek12/offensive-identification.

The paper is organized as follows: Section 2 introduces the related works. Section 3 describes the data set and the preprocessing methods that we used. Section 4 describes our system architecture and training strategy. Experimental results are presented in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work

One of the most popular and successful methods in last year's OffensEval challenge (Zampieri et al., 2019b) is transfer learning. Recently, transfer learning in NLP using transformer-like architecture has significantly improved on the state-of-the-art in natural language understanding. Despite their success on the variety of NLP benchmarks, such pre-trained models might fail to generalize to natural language tasks from a different distribution. BERT model pre-trained on specific domain data set presented a better performance compares to the model trained on Wikipedia corpus, such as SciBERT (Beltagy et al., 2019) and BioBERT (Lee et al., 2019).

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2111–2116
Barcelona, Spain (Online), December 12, 2020.

	Class	Train		Dev
A	NOT	50,000	1,500,000 unlabeled	620
	OFF	50,000		240
B	TIN	10,000	82,000 unlabeled	213
	UNT	10,000		27

Table 1: Data Distribution: Train data is a subset of training set provided by the task organizers, only the most confident examples are assigned *hard-label* (NOT/OFF, TIN/UNT), others are treated as unlabeled data. We only use about 1/6 of the provided data from sub-task A and 1/2 from sub-task B to develop our system. Development set is the test set from last year competition.

In BERT paper (Devlin et al., 2019), the authors suggested using the output of the [CLS] token for classification. However, some researches showed that adding other layers like CNN (Rozental and Biton, 2019) or RNN (Mozafari et al., 2019) on top of BERT embedding also improves the classification result.

For a supervised learning task, a labeled data set is required to train the model. However, the amount of labeled data is minimal. To improve the accuracy and robustness of the model, using a teacher-student training process was a successful approach used in ImageNet training, which called Noisy Student. In this paper, we apply this approach to train our BERT model on a large-scale of semi-labeled tweets data.

3 Data and Preprocessing

3.1 Data Description

The OffensEval 2020 - English language track are divided into three sub-tasks:

A - Offensive language identification

B - Categorization of offense types

C - Target identification *(Not attend)*

In sub-task A, we predict if the post is *Offensive (OFF)* - Containing offensive language or a targeted offense; or *Not Offensive (NOT)* - No offensive language or profanity. In sub-task B, we classify the offenses into two types: *Targeted Insult and Threats (TIN)* - Containing an insult or a threat to an individual, a group, or others; or *Untargeted (UNT)* - Containing non-targeted profanity and swearing.

The public training data for this task is more than 9 million sentences of tweets for sub-task A and nearly 190 thousand sentences for sub-task B. However, there is no human label provided. Multiple supervised models were used to score those sentences. Each sentence is given along with the average of predicted confidences (*AVG_CONF*) and the confidences' standard deviation (*CONF_STD*).

An important element for the Noisy Student training method to work well is that the teacher model should be trained on clean labels. Therefore, to limit the noises of the given data, we only select the sentences that have low standard deviation with the average confident scores closed to 1 (for positive class) or closed to 0 (for negative class). A subset from the remaining data is treated as unlabeled data to use in training student models. We do not use all provided sentences due to time and computational limitations.

As suggested by the organizers, we use the public data set from the last year's competition (Zampieri et al., 2019a) to evaluate the model. Details of the data set are showed in Table 1.

3.2 Data Preprocessing

On social media, people prefer to use emoji and hashtags to show their expressions. Therefore, similar to Liu et al. (2019), we convert emoji[1] and hashtag[2] to English words to maintain their semantic meanings.

Another common syntax that can be found on Twitter's posts is micro-text, which might also contain offensive meaning (eg. 'af' - 'as fuck', 'kys' - 'kill your self', etc.). A list of microtext[3] from Satapathy et al. (2019) was used to normalize those words.

[1] https://github.com/carpedm20/emoji

[2] https://github.com/grantjenks/python-wordsegment

[3] https://github.com/npuliyang/Microtext_Normalization/

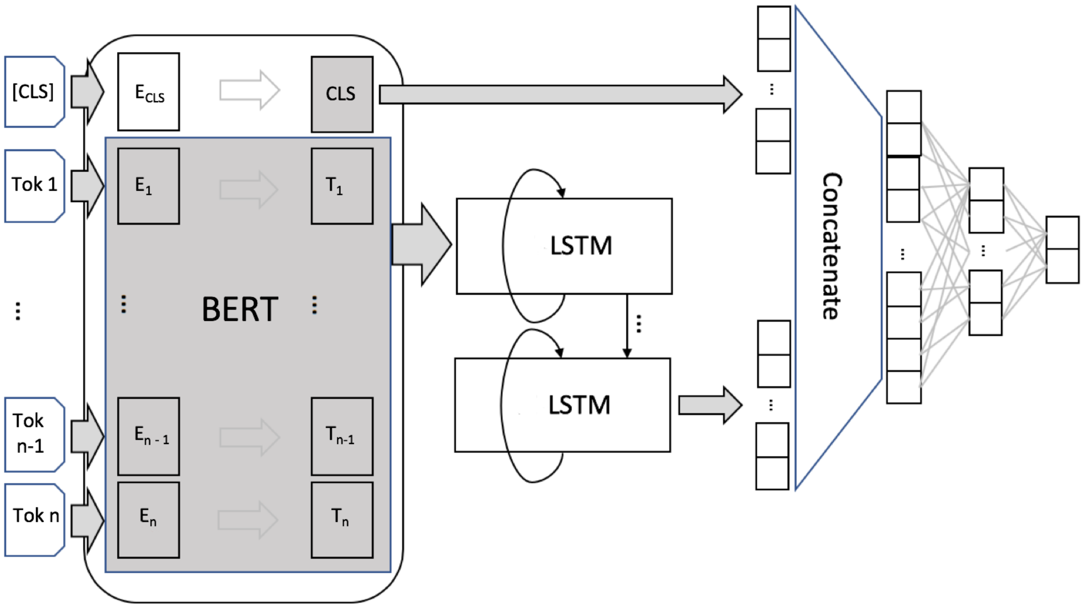

Figure 1: Architecture of BERT-LSTM model

We convert all text to lowercase and remove special characters as well.

4 System Description

4.1 Pretrained BERT with Tweets data

Due to the limitation of computational power, we decide not to pre-train BERT model from scratch but fine-tune from the *BERT-Large, Uncased (Whole Word Masking)* checkpoint.

In BERT's vocabulary, there are 994 tokens marked as *'unused'*. These tokens are suggested to be used to expand the vocabulary. In our case, we only replace 150 of them with the top occurrences and offensive-related words from the training set. We then use those tweet sentences to pre-train this BERT model. We follow the instruction of pre-training model from Google BERT github[4]. However, since tweets data are single short sentences, we modify the processing and training script to remove the *Next Sentence Prediction* loss and only perform the *Masked LM* task.

The checkpoint we choose to train our Offensive Identifying classifier has:

$$masked\_lm\_accuracy = 0.667$$

$$masked\_lm\_loss = 1.749$$

Finally, we use the Transformers library from HuggingFace (Wolf et al., 2019) to convert the Tensorflow checkpoint to Pytorch and perform later training process.

4.2 BERT-LSTM model

In our approach, we take the output vectors of all the word tokens. Those tokens are sent through LSTM layers, then concatenated with the [CLS] token and finally passed to a fully connected neural network to perform the final classification (Figure 1).

System	Macro-F1
BERT-Large, Uncased	73.4
Tweet's BERT-Large	74.7
Tweet's BERT-LSTM	75.3
Noisy Student Tweet's BERT-LSTM	**77.0**

Table 2: Result of sub-task B on the development set with different training setup: BERT-Large, Uncased, Tweet's BERT-Large, Tweet's BERT-LSTM and Noisy Student Tweet's BERT-LSTM

4.3 Noisy Student training

Although there are a large number of tweet sentences provided as a training set, the labels are predicted by other supervised models, many of them have low confidences with high standard deviations. To leverage this enormous data, we use the Noisy Student training method, which was successfully applied to train the current state-of-the-art of ImageNet challenge.

We only select the most confident instances from the training set and assign hard-label (NOT/OFF, TIN/UNT) with the threshold of 0.5. These instances are used to train the 'Teacher' model.

Then we split the unlabeled data set to multiple subsets. At each iteration, we use the 'Teacher' model to score one subset to generate the pseudo labels. The 'Student' model is then trained on both the teacher's training data and the new subset with those pseudo labels. Finally, we iterate the process by putting back the student as a teacher to generate pseudo labels on a new subset and train a new student again.

To learn the first 'Teacher' model, we minimize the Cross-Entropy loss on hard-labeled data. Then both soft and hard pseudo labels are generated for unlabeled data to train the 'Student' model. A soft label is a discrete probability output of a the network given by

$$\tilde{y}_{soft(i)} = softmax(z)_i = \frac{exp(z_i)}{\sum_j exp(z_j))} \tag{1}$$

where i denotes the i^{th} class and z denotes the logits of the network. Then a hard label is assigned by

$$\tilde{y}_{hard} = argmax(\tilde{y}_{soft}) \tag{2}$$

We use the combined objective of Cross-Entropy loss ($\mathcal{L}_{CE}$) on hard labels and Kullback-Leibler Divergence loss ($\mathcal{L}_{KLDiv}$) on soft labels with soft-label ratio $\alpha = 0.3$ to train the 'Student' model as in (3).

$$loss = (1 - \alpha) * \mathcal{L}_{CE} + \alpha * \mathcal{L}_{KLDiv} \tag{3}$$

According to Xie et al. (Xie et al., 2019), a larger student model with added noise will force the model to learn harder, hence improves its performance. In our implementation, we increase the number of Fully Connected layers and add Dropout layers with the probability range from 0.3 to 0.5 throughout the training process to achieve that.

5 Result

The official evaluation metric for both sub-task A and B in the competition is Macro-F1. Since the data set of sub-task B is much smaller, during the development phase, we use it to conduct experiments and compare the results of different training setups.

The results are evaluated on OLID's test set with the same training hyper-parameters for all setups (e.g learning rate, batch size). The Noisy Student model is trained as described in previous section with three iterations (3 teacher models). The last student model is used to generate final submission. To train other baseline models, we choose the confidence threshold of 0.5 to assign hard labels on the given training set.

[4] https://github.com/google-research/bert

System	Macro-F1
Noisy Student Tweet's BERT-LSTM	81.3

Table 3: Result of sub-task A on the development set using Noisy Student Tweet's BERT-LSTM

System	Macro-F1
Sub-task A	90.59
Sub-task B	73.62

Table 4: Result on official test set

Based on the results of validation (Table 2), we choose to use the Noisy Student Tweet's BERT-LSTM as our selected model for the final submission.

For sub-task A, we train the model with the Noisy student method only and report the result in Table 3.

In Table 4, we report the result on the official test set on CodaLab [5]. For sub-task A, we are ranked 55th with the F1 score of 90.59. However, it's only 1.6 points lower than the first system. We suppose that it is because we only used 1/6 of provided data to train the model. We did not achieve all the potential of the training method. In sub-task B, our approach performed 4.5 points better than the next system. By using the Noisy Student training method, our model can leverage the enormous amount of data despite the noisy labels, hence improve the performance.

6 Conclusion

In this paper, we have described the system that we use to attend to the SemEval-2020 competition - Task 12, which reaches second place at sub-task B of English language track. By updating the vocabulary and fine-tuning the BERT model from the existing checkpoint, we can quickly adapt the pre-trained model to a new domain (Tweets). We also extend the BERT classifier by LSTM layers and use the Noisy Student training approach to improve the accuracy and robustness of the models without human annotation required.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09.

Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Jamshid Mozafari, Afsaneh Fatemi, and Mohammad Ali Nematbakhsh. 2019. Bas: An answer selection method using bert language model. *arXiv preprint arXiv:1911.01528*.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A Large-Scale Weakly Supervised Dataset for Offensive Language Identification. In *arxiv*.

[5] https://competitions.codalab.org/competitions/23285

Alon Rozental and Dadi Biton. 2019. Amobee at SemEval-2019 tasks 5 and 6: Multiple choice CNN over contextual embedding. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 377–381, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Ranjan Satapathy, Yang Li, Sandro Cavallari, and Erik Cambria. 2019. Seq2seq deep learning models for microtext normalization. pages 1–8, 07.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Qizhe Xie, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. Self-training with noisy student improves imagenet classification. *CoRR*, abs/1911.04252.

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. pages 1415–1420, 01.

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*, pages 75–86, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.