NULI at SemEval-2019 Task 6: Transfer Learning for Offensive Language Detection using Bidirectional Transformers

Ping Liu
Department of Computer Science
Illinois Institute of Technology
pliu19@hawk.iit.edu

Wen Li
Department of Linguistics
Indiana University
wl9@indiana.edu

Liang Zou
Department of Mathematics
New York University
lz1904@nyu.edu

Abstract

Transfer learning and domain adaptive learning have been applied to various fields including computer vision (e.g., image recognition) and natural language processing (e.g., text classification). One of the benefits of transfer learning is to learn effectively and efficiently from limited labeled data with a pre-trained model. In the shared task of identifying and categorizing offensive language in social media, we preprocess the dataset according to the language behaviors on social media, and then adapt and fine-tune the Bidirectional Encoder Representation from Transformer (BERT) pre-trained by Google AI Language team[1]. Our team NULI wins **the first place (1st)** in Sub-task A - Offensive Language Identification and is ranked 4th and 18th in Sub-task B - Automatic Categorization of Offense Types and Sub-task C - Offense Target Identification respectively.

1 Introduction

Anti-social online behaviors, including cyberbullying, trolling and offensive language (Xu et al., 2012; Kwok and Wang, 2013; Cheng et al., 2017), are attracting more attention on different social networks. The intervention of such behaviors should be taken at the earliest opportunity. Automatic offensive language detection using machine learning algorithms becomes one solution to identifying such hostility and has shown promising performance.

In SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (Zampieri et al., 2019b), the organizers collected tweets through Twitter API and annotated them hierarchically regarding offensive language, offense type, and offense target. The task is divided into three sub-tasks: a) detecting if a post is offensive (OFF) or not (NOT); b) identifying the offense type of an offensive post as targeted insult (TIN), targeted threat (TTH), or untargeted (UNT); c) for a post labeled as TIN/TTH in sub-task B, identifying the target of offense as individual (IND), group of people (GRP), organization or entity (ORG), or other (OTH). The three sub-tasks are independently evaluated by macro-F1 metric.

The challenges of this shared task include: a) comparatively small dataset makes it hard to train complex models; b) the characteristics of language on social media pose difficulties such as out-of-vocabulary words and ungrammatical sentences; c) the distribution of target classes is imbalanced and inconsistent between training and test data. To address the problem of out-of-vocabulary words especially emoji and hashtags, we preprocess each tweet by interpreting emoji as meaningful English phrases and segmenting hashtags into space separated words. The classifiers we experiment with include: linear model with features of word unigrams, word2vec, and Hatebase; word-based Long Short-Term Memory (LSTM); fine-tuned Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2018). We choose BERT for our official submission, since it performs the best in our experiments.

In the rest of this paper, we organize the content as follows: related work of hostility on social media is stated in section 2; section 3 introduces data description, details of preprocessing, and the methodology of our models; experimental results are discussed in section 4. We also present the conclusion of our work at the end of paper.

2 Related Work

Schmidt and Wiegand (2017) surveyed features widely used for hate speech detection, including simple surface feature, word generalization,

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

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 87–91
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

knowledge-based features, etc. Davidson et al. (2017) reported hate speech detection results using word *n*-grams and sentiment lexicon and provided insights on misclassified examples. A proposal of typology of abusive language sub-tasks is presented in (Waseem et al., 2017). (Liu et al., 2018) also discuss that the forecasting of the future hostility on Instagram can be divided into two levels: presence and intensity. In addition to English, researchers also investigated offensive language detection for Chinese (Su et al., 2017) and Slovene (Fišer et al., 2017). In the shared task on aggression identification organised as part of the first workshop on trolling, aggression and cyberbullying (TRAC - 1) at COLING 2018, word/character *n*-grams and word embeddings were the most commonly used features among the participants, and the most popular classifiers were SVM, LSTM, and RNN. The best performing system employed bidirectional LSTM on Glove embeddings.

3 Data and Methodology

3.1 Data Description

Offensive Language Identification Dataset (OLID) (Zampieri et al., 2019a) is collected from Twitter API by searching certain keywords set. The keywords include some unbiased targeted phrase such as ‘she is’, ‘he is’ and ‘you are’ which have high proportional offensive tweets. The distribution of offensive tweets is controlled around 30% by using different sampling methods. Another observation reported in the paper is political tweets tend to be more likely offensive using keywords as ‘MEGA’, ‘liberal’ and ‘conservative’.

The main task of this competition is decomposed into three different levels according to the hierarchical annotation: a) Offensive Language Detection b) Categorization of Offensive Language c) Offensive Language Target Identification. All the three different tasks share the same dataset, and the latter one is the subset of the previous one.

The tasks release the dataset into three different parts, which are the startingKit, training dataset and testing dataset. The summary of dataset distribution is concluded in the Table1. From the table, it is easy to observe that the distribution of three splittings is a little twisted which should be expected in real life, and also make the tasks much harder.

Class	StartKit	Training	Testing
NOT	243	8840	620
OFF	77	4400	280
TIN	38	3876	213
UNT	39	524	27
IND	30	2407	100
GRP	7	1074	78
OTH	2	395	35

Table 1: Data Distribution: The first two rows are the class distribution of sub-task A. The mid part two rows are the class distribution of sub-task B. The last three rows are the class distribution of sub-task C.

3.2 Preprocessing

Emoji substitution We use one online emoji project on github [2] which could map the emoji unicode to substituted phrase. We treat such phrases into regular English phrase thus it could maintain their semantic meanings, especially when the dataset size is limited.

HashTag segmentation The HashTag becomes a popular culture cross multi social networks, including Twitter, Instagram, Facebook etc. In order to detect whether the HashTag contains profanity words, we apply word segmentation using one open source on the github [3]. One typical example would be ‘#LunaticLeft’ is segmented as ‘Lunatic Left’ which is obviously offensive in this case.

Misc. We also convert all the text into lower case. ‘URL’ is substituted by ‘http’, since ‘URL’ does not have embedding representation in some pre-trained embedding and models. Consecutive ‘@USER’s are limited to three times to reduce the redundancy.

3.3 Methodology

Linear model We firstly select Logistic Regression as our baseline model to determine the lower bound performance that we should compare. First we cross-validate hyper-parameters of different vectorizers to build bag of words representation. Secondly, we adopt the pre-trained word2vec model from google [4], then aggregate the maximum and average value in each dimension.

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

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

[4] https://code.google.com/archive/p/word2vec/

(a) Sub-task A

System	MacroF	Acc
All NOT	0.4004	0.6677
All OFF	0.2494	0.3323
Linear	0.7102	0.7273
LSTM	0.7166	0.7659
BERT	**0.7826**	0.8485

(b) Sub-task B

System	MacroF	Acc
All TIN	0.4686	0.8818
All UNT	0.1057	0.1182
Linear	**0.6028**	0.8000
LSTM	0.5029	0.8795
BERT	0.3830	0.8682

(c) Sub-task C

System	MacroF	Acc
All GRP	0.1441	0.2758
All IND	0.2554	0.6211
All OTH	0.0623	0.1031
Linear	0.5607	0.7062
LSTM	0.5056	0.7036
BERT	**0.8435**	0.7294

Table 2: Results on Dev Data.

Thirdly, we use the dictionary Hatebase API[5] to aggregate the hate words in each category. We validate all the features combinations, then report the accuracy and F1 with the highest to determine the model parameters.

LSTM Long Short-Term Memory is introduced in 1991 (Hochreiter and Schmidhuber, 1997) which is an more powerful extension of recurrent neural network. The gates inside of LSTM could prevent gradient vanishing problem, to memorize the long time dependency. LSTM has been used in tons of natural language processing task, such as sentiment classification, neural translation, language generation etc. We would also like to use LSTM as our second powerful baseline model to compare and report the result. The specific setting is the following: the input is mapped from one-hot encoder into a shared embedding layers with dimension 140; the hidden units of LSTM is 64 and follower by a dropout layer with rate 0.5. The maximum sequence length is 140, thus the sentences would be either cut off or padded.

BERT Google research team releases Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2018) and achieve state of the art results on many NLP tasks. BERT uses identical multi-head transformer structure that is introduced in (Vaswani et al., 2017). The model is pre-trained on huge corpus from different sources. Since the dataset size in this SemEval-2019 Task 6 is not that big, we pass the dataset into the pre-trained BERT model, and report the loss and accuracy at each epoch. The observation from experiments shows that after 1st or 2nd epochs, the model converges fast and always get very lower loss on the validation set. In such case, in the sub-task B and sub-task C, we report the macro-F1 score after the model trains after 1st, 2nd and 3rd epoch.

4 Experiment Results

The evaluation metric of this task is Macro-F1, which is the unweighted-average F1 of all the classes. The imbalance distribution makes the macro-F1 hard to achieve, and usually the score is penalized by the minority class. Weighted-loss is one solution during the training time to balance the model not to lead to the majority class prediction.

In the table 2 and 3, we report the results of our dev-dataset and final test dataset. From the table 2, we list the performance of our three selected models for each sub-task. The data is stratified split into 9:1 as train and test. There is also one independent validation set to determine the model selection that is split from train set. One observation from the table shows the problem of imbalanced data, so that higher accuracy does not guarantee higher macro-F1 score. Thus the stop criterion is based on average loss of validation set we mentioned before. Based on the results of validation, we choose to use BERT as our selected model for the final submission.

In the table 3, it shows the results on official test dataset. It should be noticed that in the sub-task A, we also submit one result of a Bagging classifier with number 50, and Logistic Regression is the weak classifier. The features are the same with linear model we mentioned before. The result from BERT model sub-task A achieves the 1st place among all the participants. BERT-3 denotes we train BERT with only 3 epochs. Same notation with the latter two sub-tables. In the sub-task B and sub-task C, the results are not as good as sub-task A due to two reasons: 1) the class distribution is more skewed than that of sub-task A. 2) the number of training instance is much smaller than sub-task A. The worst performance is sub-task C,

[5] http://www.hatebase.org

(a) Sub-task A

System	MacroF	Acc
All NOT	0.4189	0.7209
All OFF	0.2182	0.2790
Bagg	0.7558	0.8105
Linear	0.7501	0.7953
BERT-3	**0.8286**	**0.8628**

(b) Sub-task B

System	MacroF	Acc
All TIN	0.4702	0.8875
All UNT	0.1011	0.1125
BERT-1	0.6932	0.8875
BERT-2	0.4702	0.8875
BERT-3	**0.7159**	**0.8958**

(c) Sub-task C

System	MacroF	Acc
All GRP	0.1787	0.3662
All IND	0.2130	0.4695
All OTH	0.0941	0.1643
BERT-1	0.5267	**0.7277**
BERT-2	**0.5598**	0.6948

Table 3: Results on Test Data.

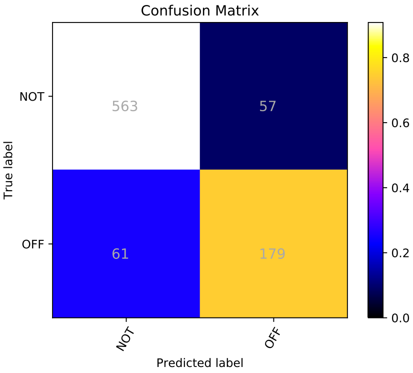

Figure 1: Sub-task A, BERT model after fine-tuning

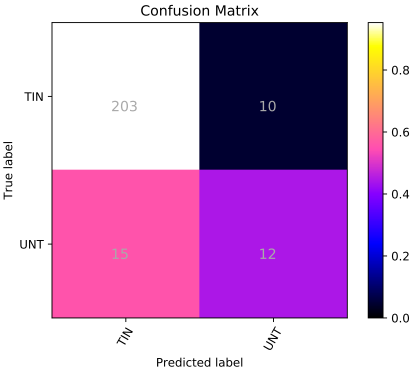

Figure 2: Sub-task B, BERT model after fine-tuning

since it is three-class classification, and the 'OTH' class has very few examples.

The confusion matrix of three sub-tasks are shown in fig 1, 2, and 3. This is another way to explain the results as we discussed before. The figures are provided by the organizers, and we use the figures to summarize test distribution in the table 1. In the previous section, we mentioned the discrepancy of class distribution between training and test datasets. For example, in sub-task C, the class 'OTH' constitutes 0.101 of the training data, while it makes up 0.164 of the test data. This adds difficulty to the task, however, we are often confronted with the same situation in real-world problems.

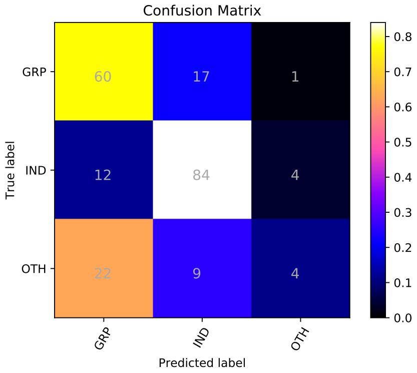

Figure 3: Sub-task C, BERT model after fine-tuning

5 Conclusion

Offensive language and online hostility is crucial on the social network. The minority proportion of the nature and morphological language are the difficulties to achieve high performance. The Diversity and evolution of the language at different ages is another challenge for social media detection task. As a conclusion, our work shows the competitive results in this shared task using customized processing to dataset, as well as the power of pre-trained model. In real life, labeled data is always limited and requires expensive human labors. In such case, transfer learning is always a good option to get started. Domain adaption also has prior knowledge of specific domain before doing any modeling work on hand. How to tune the parameters is nontrivial, and there are a lot of more efficient ways to be explored, which could yield better performance.

References

Justin Cheng, Michael Bernstein, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. 2017. Anyone can become a troll: Causes of trolling behavior in online discussions. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, pages 1217–1230. ACM.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of ICWSM*.

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

Darja Fišer, Tomaž Erjavec, and Nikola Ljubešić. 2017. Legal Framework, Dataset and Annotation Schema for Socially Unacceptable On-line Discourse Practices in Slovene. In *Proceedings of the Workshop Workshop on Abusive Language Online (ALW)*, Vancouver, Canada.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting Tweets Against Blacks. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Ping Liu, Joshua Guberman, Libby Hemphill, and Aron Culotta. 2018. Forecasting the presence and intensity of hostility on instagram using linguistic and social features. In *Twelfth International AAAI Conference on Web and Social Media*.

Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection Using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics*, pages 1–10, Valencia, Spain.

Huei-Po Su, Chen-Jie Huang, Hao-Tsung Chang, and Chuan-Jie Lin. 2017. Rephrasing Profanity in Chinese Text. In *Proceedings of the Workshop Workshop on Abusive Language Online (ALW)*, Vancouver, Canada.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In *Proceedings of the First Workshop on Abusive Langauge Online*.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 656–666. Association for Computational Linguistics.

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)*.