jhan014 at SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media

Jiahui Han
University of Ottawa
jhan014@uottawa.ca

Shengtan Wu
Jackson State University
shengtan.wu@students.jsums.edu

Xinyu Liu
Purdue University
liu1957@purdue.edu

Abstract

In this paper, the team jhan014 presents two methods to identify and categorize the offensive language in Twitter. In the first method, we develop a deep neural network consisting of bidirectional recurrent layers with Gated Recurrent Unit (GRU) cells and fully connected layers. In the second method, we establish a probabilistic model, modified sentence offensiveness calculation (MSOC) to evaluate the sentence offensiveness level and target level according to different sub-tasks. Based on task results, We evaluate the performance of each method based on F1 score and analyze the advantages and disadvantages of these two methods with the type I error and type II error. In conclusion, deep neural network behaves well in all subtasks but has more type I error and fails to categorize subclasses with minor data or less character, while MSOC model does better in target categorizing but has more type II error in offensive identifying.

1 Introduction

With the popularity of social media like Twitter, offensive language has become a serious problem(Zampieri et al., 2019b) on these media platforms. People have to face with abusive behavior from others in social media from time to time. To solve this problem, finding a method to identify and categorize offensive languages is an urgent need.

In this paper, two different methods, deep learning method and modified sentence offensiveness calculation method, are used to categorize the type and target of offensive language and the difference of results are revealed and analyzed.

2 Related Work

Deep learning method: Deep learning methods are widely used in natural language processing (Liu et al., 2016). Models like Recursive neural network are commonly used to identify if a sentence contain certain emotion. In our work, a deep neural network with GRU layers and all connection layers is built.

Offensiveness Content Filtering: Offensive language targets can be understood through the sentence structure (Silva et al., 2016) or lexical analysis (ElSherief et al., 2018). We take both sentence structure and the offensiveness level of words into consideration. Furthermore, we also concentrate on the special punctuation (like @ and #) in online social media.

3 Methodology and Data

3.1 Deep Neural Network

In the offensive language detection task, we developed a deep neural network based system with binary cross-entropy output.

System Design The system consists of bidirectional recurrent layers with Gated Recurrent Unit(GRU) cells and fully connected layers (Chung et al., 2014). Because the output of the last time-step is used as the embedding of a sentence, we conduct zero padding in the beginning of each sequence to construct the feature matrix. The system architecture is shown in Table 1.

Optimization Steps Parameters in both RNN layers and Dense layers are initialized by Xiaver initialization method (Glorot and Bengio, 2010). The model is optimized by Adam optimization method with 0.01 learning rate. While training the

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 652–656
Minneapolis, Minnesota, USA, June 6–7, 2019.

Layer Name	Output Dimension	Parameter #
Embedding	100	1000000
GRU	128	63360
GRU	128	74112
GRU	128	74112
GRU	128	74112
Dense	256	33024
Dense	128	32896
Dense	64	8256
Dense	32	2080
Dense	2	66

Table 1: System Architecture

neural network, an early stopping method with 2-iteration tolerance is applied to monitor the process. Once the early stopping method is triggered, we manually lower the learning rate by $1/10$ to overcome the vibration and search for a smaller minimum loss.

3.2 Modified Sentence Offensiveness Calculation

Based on the sentence offensiveness calculation method in this paper(Chen et al., 2012), we develop a model to evaluate the sentence offensiveness.

Offensiveness Dictionary Construction We can always find pejoratives, profanities, or obscenities in offensive twitters. Strongly profanities are always undoubtedly offensive when at users or related to some topics (like #) directly; but there are many other weakly pejoratives and obscenities that may also be offensive.
Word offensiveness is defined(Chen et al., 2012) as: for each offensive word, w, its offensiveness

$$O_w = \begin{cases} a_1 & \text{if w is a strongly offensive word} \\ a_2 & \text{if w is a weakly offensive word} \\ 0 & \text{otherwise} \end{cases}$$

where $0 < a_1 < a_2 < 1$, for the offensiveness of strongly offensive words is higher than weakly offensive words.

Syntactic Intensifier Detection We also built the syntactic features by an intensifier(Zhang et al., 2009). In a sentence, words syntactically related to offensive word, w, are categorized in an intensifier set, $i_w = \{c_1, \ldots, c_k\}$, for each word c_j, its intensify value, d_j, is defined as

$$d_j = \begin{cases} b_1 & \text{if } \mathrm{c_j} \text{ is @ or \#} \\ b_2 & \text{if } \mathrm{c_j} \text{ is an offensive word} \\ 1 & \text{otherwise} \end{cases}$$

where $0 < b_1 < b_2 < 1$, for offensive words used to describe users are more offensive than the words used to describe other offensive words. Thus, the value of intensifier, I_w, for offensive word, w, can be calculated as $\sum_{j=1}^{k} d_j$.

Sentence Level Offensiveness Value Consequently, the offensiveness value of sentence, s, becomes a determined linear combination of words' offensiveness

$$O_s = \sum O_w I_w$$

From the training data, we learn two thresholds θ_1, θ_2. For each sentence, s, we apply these two values

$$P(s = OFF) = \begin{cases} 1 & \text{if } \mathrm{O_s} > \theta_2 \\ \frac{O_s - \theta_1}{\theta_2 - \theta_1} & \text{if } \theta_1 \leq \mathrm{O_s} \leq \theta_2 \\ 0 & \text{if } \mathrm{O_s} < \theta_1 \end{cases}$$

If the offensiveness value is greater than θ_1, the language will be seen as offensive, while if it is smaller than θ_2 then the language will be not offensive. Otherwise, the result will follow a probabilistic distribution.

When solving other sub-tasks, this method can also be used with changing the dictionary and redefine the target words list.

3.3 Data

We use the datasets in Zampieri et al. (2019a) and apply following methods to preprocess or transform the data.

3.3.1 Preprocessing

The raw twitter data is preprocessed by a data pipeline. All the information which has nothing to do with word vectors such as stop words and emojis are stripped and the output of the pipeline are lower-case stemmed word sequencies.

3.3.2 Word Embedding

A word embedding step is applied to transform the text into numerics for deep neural networks. 100-dimensional Global Vectors(GloVe) word embeddings trained with twitter data are applied in this study considering the trade-off between performance and efficiency of the training process (Pennington et al., 2014). We also explore embedding layers in this study and the pretrained embedding out-performs the embedding layer because of

the immense amount of information brought by GloVe's training set with 27-billion tweets.

4 Results

4.1 Sub-task A - Offensive language identification

When identifying whether a sentence is offensive or not, two methods show great difference while the accuracy and F1-score are close (see Table 2). In RNN method, there is more type I error (see Figure 1) which means the model classifies some non-offensive sentences as offensive ones. Since origin dataset is unbalanced, the neural network may not have enough non-offensive training examples to learn. Consequently, it cannot catch the feature and structure of the non-offensive sentences.

In MSOC method, this problem is improved. Due to fixed human defined offensiveness dictionary, the non-offensive sentence is not easily misclassified as offensive one. However, since there are still some offensive words appeared in dataset that are not defined in the dictionary, there is still much type II error (see Figure 2).

System	F1 (macro)	Accuracy
All NOT	0.4189	0.7209
All OFF	0.2182	0.2790
RNN	0.6899	0.7395
MSOC	0.6761	0.7895

Table 2: Results for Sub-task A.

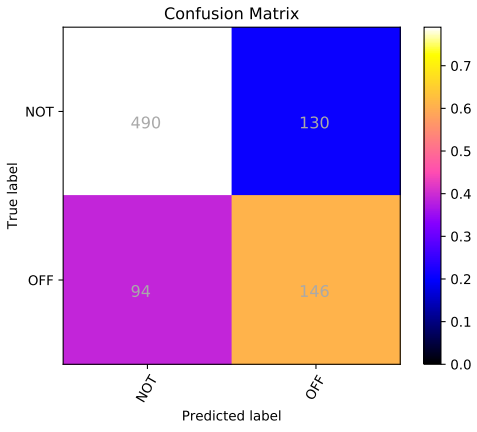

Figure 1: Sub-task A,RNN method

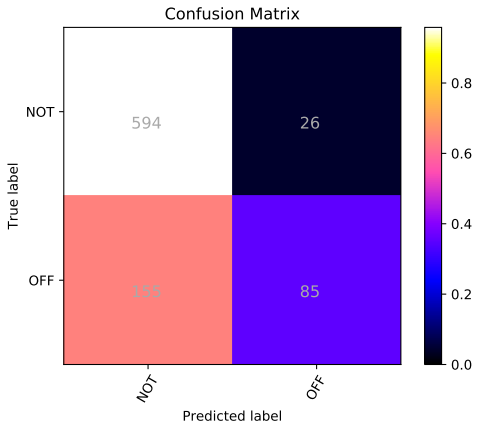

Figure 2: Sub-task A,MSOC method

4.2 Sub-task B - Automatic categorization of offense types

The behavior of MSOC method defeats RNN method from all aspects (see Table 3 and Figure 3, 4) when categorizing the types of offense. This is because usually targeted offensive language have different sentence structure with untargetted ones, this make it a really high accuracy approach to categorize offensive type. In details, a target sentence always contains third-person pronouns like him her it them. And in most target tweets, the sentence has some special punctuation like @ and also related to some hot topics #.

System	F1 (macro)	Accuracy
All TIN	0.4702	0.8875
All UNT	0.1011	0.1125
RNN	0.6153	0.8667
MSOC	0.7545	0.925

Table 3: Results for Sub-task B.

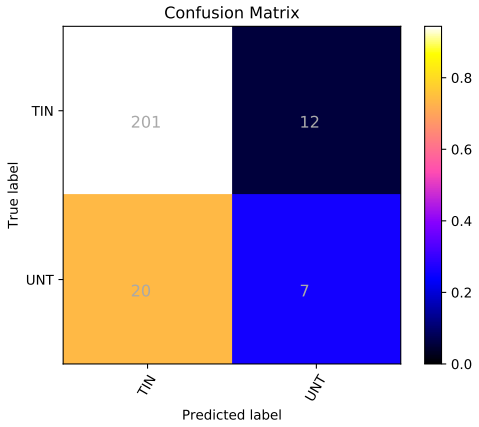

Figure 3: Sub-task B,RNN method

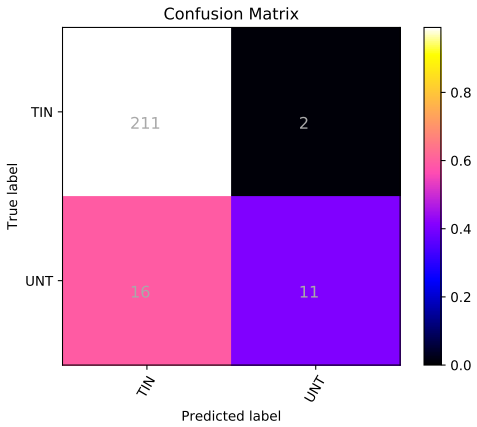

Figure 4: Sub-task B,MSOC method

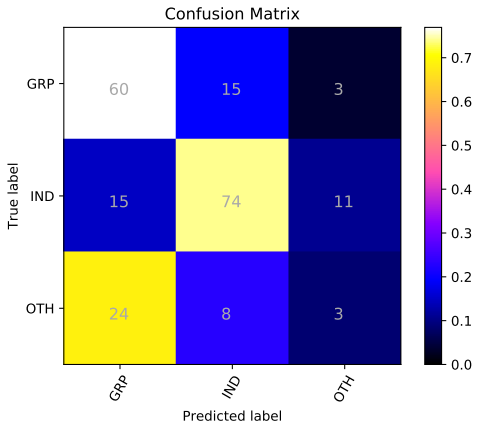

Figure 6: Sub-task C,MSOC method

4.3 Sub-task C - Offense target identification

In offense target identification, RNN method, although has the similar accuracy and F1 score with MSOC method (see Table 4), fails to classify any of the test sentences into 'OTH' class.(see Figure 5) The main reason of this result is 'OTH' class is not as characteristic as other two classes and the partition of this class is the smallest as well. On contrast, MSOC method can successfully classify some test sentences in 'OTH' class. (see Figure 6) This may contribute to the predefined dictionary and sentence structure.

System	F1 (macro)	Accuracy
All GRP	0.1787	0.3662
All IND	0.2130	0.4695
All OTH	0.0941	0.1643
MSOC	0.5149	0.6432
RNN	0.4630	0.6432

Table 4: Results for Sub-task C.

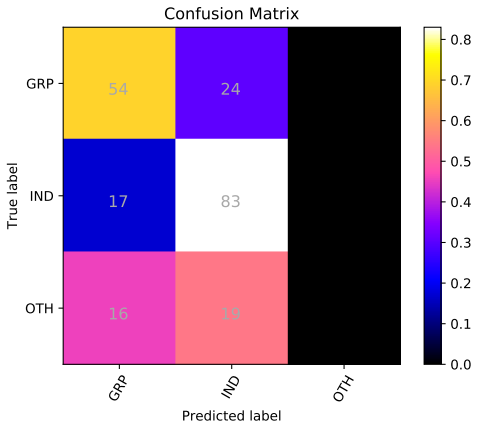

Figure 5: Sub-task C,RNN method

5 Conclusion

RNN is an easy-implemented and high-efficiency method to solve classification problem in natural language processing. In this case, RNN shows an acceptable result but it has many obvious drawbacks. Such as high recall rate when handling unbalanced data, fail to classify certain class if the class is lack of obvious character. The MSOC mehod, on the contrary, can give classification result of same quality. Even though MSOC cannot improve the accuracy or the F1 score of classification to a great extend, we believe we can combine this method with deep learning method to get a better result in similar problems in the future.

References

Y. Chen, Y. Zhou, S. Zhu, and H. Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 71–80.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding. 2018. Hate Lingo: A Target-based Linguistic Analysis of Hate Speech in Social Media. *arXiv preprint arXiv:1804.04257*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *CoRR*, abs/1605.05101.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP*.

Leandro Araújo Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. *CoRR*, abs/1603.07709.

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)*.

Changli Zhang, Daniel Zeng, Jiexun Li, Fei-Yue Wang, and Wanli Zuo. 2009. Sentiment analysis of chinese documents: From sentence to document level. *Journal of the American Society for Information Science and Technology*, 60(12):2474–2487.