jhan014 at SemEval-2019 Task 6: A Comparative Study: Deep Neural Networks and Modified Sentence Offensiveness Calculation for Offensive Language Detection

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

Xingyu Liu
Purdue University
liu1957@purdue.edu

Jiahui Han
University of Ottawa
jhan014@uottawa.ca

Abstract

In this study, we (**team name: jhan014**) apply two different methods for offensive language detection tasks(Pennington et al., 2014). One of the methods using a target-based probabilistic model with linguistic and psycholinguistic analysis (ElSherief et al., 2018). The other method is a Deep Neural Network(DNN) based system built with a novel optimization strategy. Our result analysis shows that the target-based probabilistic model outperforms DNN on specific tasks. DNN has a decent overall performance.

1 Introduction

Offensive language has become a severe social problem in the digital world. Massive offensive language may lead to multiple kinds of psychological trauma. Fortunately, machine learning technology is introduced to detect and control offensive words in social media such as Facebook and Twitter. In OffensEval competition held by Codalab(Zampieri et al., 2019b), three sub-tasks are posted:

- **Task A** is to identify if a tweet is offensive or not.
- **Task B** is to classify two types of offensive sentence: Targeted or None targeted.
- **Task C** is to classify the target of offensive word between individual, group or others.

DNN is trending in Natural Language Processing(NLP). In these studies, Recurrent Neural Network(RNN) and its variants(Gated Recurrent Unit(GRU)(Chung et al., 2014) and Long Short Term Memory(Sepp Hochreiter, 1997)) are used for sequential modeling.

The idea of transfer learning(Yosinski et al., 2014) is leveraged to map from encoded words to informative distributions is used in many NLP tasks.

Modified Sentence Offensiveness Calculation(MSOC) is a probabilistic model with multiple thresholds based on sentences structure analysis and linguistic analysis. This method has been proved efficient for large-scale hate words mining(ElSherief et al., 2018; Chen et al., 2012).

This paper presents brief description and general results of two systems using DNN and MSOC respectively. This paper also provides practical information and experience about implementation and hyper-parameter tuning. Advantages and limitations of each method are covered in the results analysis part.

2 Related Work

Offensive word detection tends to be a classification task in most cases. For DNN systems, the most widely used feature is the distributed word representations which is also referred to as *word embeddings*(Schmidt and Wiegand, 2017). The word embedding is generated according to its context environment (Mikolov et al., 2013). The algorithm projects words into a high dimensional space so that semantic similarity can be described as euclidean distance and cosine similarity between them. Word vectors are empirically used as inputs to DNN models.

RNN with front-end and back-end process has been entroduced to generate meaningful and insightful representations from documents (Zhang et al., 2018). As variants of RNN, LSTM and GRU has multiple gates in each cell to maintain hidden states so as to preserve information in long term memory. In practice, output layer of the last time step is considered a perfect representation of the whole sequence and is used in both generative models and discrimination models (Sutskever et al., 2014; Tang et al., 2015; Del Vigna12 et al., 2017; Kamkarhaghighi and Makrehchi, 2017).

Offensive language targets can be understood through the sentence structure and lexical analysis. Inspired by Named Entity Recognition(Ritter et al., 2011) and linguistic analysis(ElSherief et al., 2018). Targeted hate sentences tend to use angrier and more extreme words. The model is able to classify sentences by identifying the offensiveness level of words and tracking the target by frequently used punctuations like '@' and '#'.

3 Methodology and Data

Preprocessing steps are performed for implementations of both models. The training data set is collected from twitter and the size of training data set roughly 13000 (Zampieri et al., 2019b). Each example contains a raw tweet with punctuation and abbreviation Zampieri et al. (2019a).

The raw twitter data is preprocessed with a pipeline described as follows: All the redundent information such as stop words and emojis are stripped. In task B and task C, '#' and '@' are converted into 'hashtag' and 'at'. Stemming and tokenization are implemented for word embedding. The data is balanced with downsampling strategy on the mojority class.

3.1 Deep Neural Network

DNN based system developed for offensive detection tasks consists of GRU layers and Dense layers.

GloVe(Pennington et al., 2014) word vectors with 100-dimensions are applied in this step. The selection of dimension is a trade-off between performance and efficiency. We also explored supervised embedding layer. The layer takes one-hot encoded vectors and is set as trainable in the training process. The pre-trained word vectors outperforms the supervised embedding layer in all tasks.

Teh architecture of the system is showed as following: 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 initial learning rate(Kingma and Ba, 2014). While training the neural network, an early stopping strategy with 2-iteration tolerance is applied to monitor the process. Once the early stopping method is triggered, we manually decrease the learning rate by $1/10$ to overcome the gradient vibration. This strategy is repeated until the loss do not decrease anymore.

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

Table 1: System Architecture

3.2 Modified Sentence Offensiveness Calculation

The MSOC model evaluates sentences offensiveness and generate a probability distribution based on the features constructed by linguistic analysis.

Offensiveness Dictionary Construction Strongly offensive words such as 'f***' and 's***' are easy for our system to recognize. However, words such as 'liar' and 'stupid' may also be offensive. Thus, an offensiveness dictionary is establised following Y. Chen and his colleagues' work (Chen et al., 2012).

Word offensiveness is defined as:
for each offensive word w,

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

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 build 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, $i_w = \{c_1, ..., c_k\}$, for each word c_j, its intensify value d_j is defined as:

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

where $0 < b_1 < b_2 < 1$, for offensive words used to descrive users are more offensive than words used to descrive 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 and θ_2. For each sentence s, we apply these two values

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

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.

4 Results

4.1 Task A

In offensive identification task, both models outperforms baseline systems according to table 2. DNN has a sightly higher F1 score and MSOC has higher accuracy score. Figure1 shows that DNN model has similar performance in predicting Offensive and Non-offensive sentences. DNN takes both features and the time sequential information into consideration, yet there is still room of improvement.

4.2 Task B

In task B, table 3 shows that MSOC method outperforms all other methods. The offensiveness value O_s works as a strong evidence of targeted offensiveness. The model has a extremly high precision for targeted offensive words according to figure2. DNN model also has a reasonable accuracy score, but the F1 score is lower than MSOC.

4.3 Task C

Task C is to identify if the target of offensive language is a group, an individual or others. In this task both MSOC and DNN have similar performance on accuracy. MSOC outperforms DNN according to F1 score. Sentence level offensive value identified by offensiveness dictionary also works in target type classification problems. For our DNN model, we transform @ and # as 'at' and 'hashtag' for better target searching. DNN classifies Group and Individual with reasonable presicion, but it did not predict any samples as 'Other'. This is because word vectors show similarity in 'GRP' and 'IND'. However, word vectors may be highly random in 'OTH' class. This makes it hard to generate the decision boundary in the embedding space.

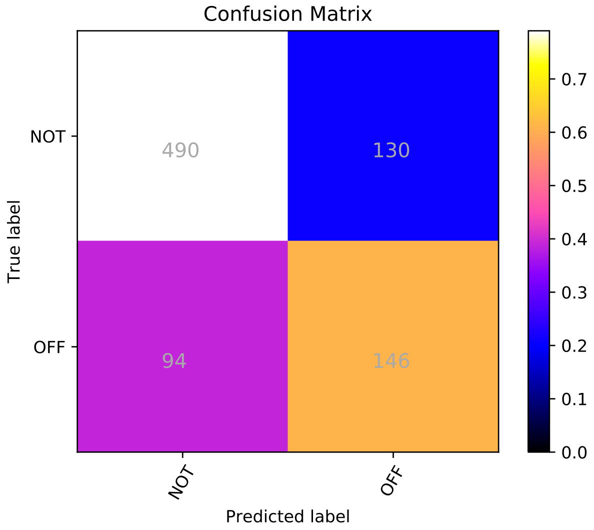

Figure 1: Sub-task A, DNN confusion matrix

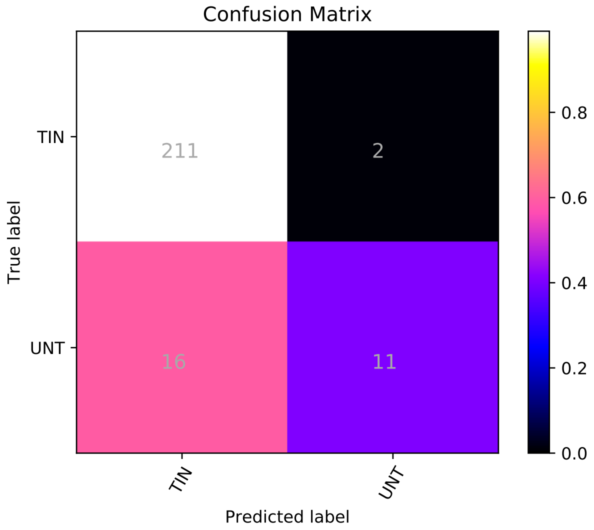

Figure 2: Sub-task B, MSOC

5 Conclusion

In this paper, we conduct comparative experiments between DNN model and MSOC model for offensive language detection. DNN model has

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

Table 2: Results for Sub-task A.

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

Table 3: Results for Sub-task B.

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
DNN	0.4630	0.6432
MSOC	**0.5149**	**0.6432**

Table 4: Results for Sub-task C.

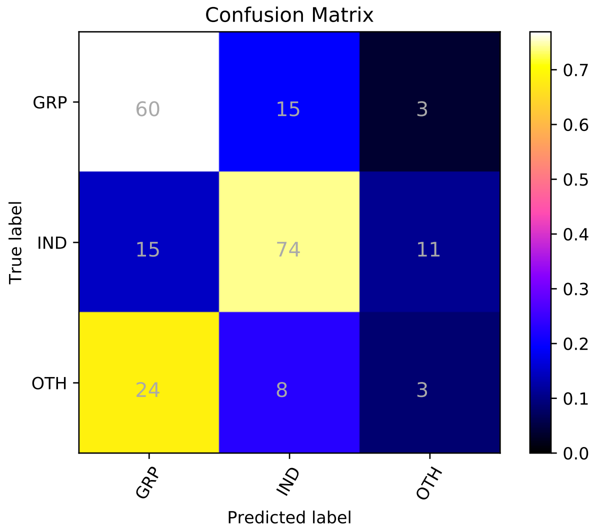

Figure 3: Sub-task C, MSOC

an acceptable overall performance and is robust comparing to methods using hand-crafted rules. However, for complicated real world problems, a straightforward training strategy can not make full use of the neural networks. To solve such problems, one DNN model may have a huge number of parameters to increase fitting power. This makes DNN models very sensitive to hyper-parameter tuning and some layers may become sparse when training. However, methods with prior knowledge,such as MSOC, perform almost flawlessly in specific tasks. Thus, our future work is breaking such problems down in the first place and applying DNN to relatively small and specific problems. This would help improving the performance of the system to another level.

References

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng 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. IEEE.

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

Fabio Del Vigna12, Andrea Cimino23, Felice DellOrletta, Marinella Petrocchi, and Maurizio Tesconi. 2017. Hate me, hate me not: Hate speech detection on facebook.

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. In *Twelfth International AAAI Conference on Web and Social Media*.

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*.

Mehran Kamkarhaghighi and Masoud Makrehchi. 2017. Content tree word embedding for document representation. *Expert Systems with Applications*, 90:241–249.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1524–1534. Association for Computational Linguistics.

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.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015. Effective lstms for target-dependent sentiment classification. *arXiv preprint arXiv:1512.01100*.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. A Hierarchical Annotation of Offensive Posts in Social Media: The Offensive Language Identification Dataset. In *arxiv preprint*.

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.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *Lecture Notes in Computer Science*. Springer Verlag.