YNU_DYX at SemEval-2019 Task 5: A Stacked BiGRU Model Based on Capsule Network in Detection of Hate

Yunxia Ding, Xiaobing Zhou[*], Xuejie Zhang
School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
yxding01@163.com, zhouxb@ynu.edu.cn, xjzhang@ynu.edu.cn

Abstract

This paper describes our system designed for SemEval 2019 Task 5 “Shared Task on Multilingual Detection of Hate”. We only participate in subtask-A in English. To address this task, we present a stacked BiGRU model based on a capsule network system. In order to convert the tweets into corresponding vector representations and input them into the neural network, we use the fastText tools to get word representations. Then, the sentence representation is enriched by stacked Bidirectional Gated Recurrent Units (BiGRUs) and used as the input of capsule network. Our system achieves an average F_1-score of 0.546 and ranks 3rd in the subtask-A in English.

1 Introduction

Hate speech is an offensive language, a statement that a person or group attacks another person or group based on characteristics such as gender, race, religion, disability, or sexual orientation. Nockleby (Nockleby, 2000) defines hate speech as “any communication that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic.” Given the huge amount of user-generated content on the Web, and in particular on social media, the problem of detecting, and therefore possibly limit the Hate Speech diffusion, is becoming fundamental, for instance for fighting against misogyny and xenophobia (Basile et al., 2019).

Microblog today has become a very popular communication tool among Internet users. Millions of users share opinions on different aspects of life everyday. And Twitter[1] is a social platform that is very popular all over the word and millions of people share their experiences, moods, attitudes toward life and discuss current issues (Pak and Paroubek, 2010). Many of the content is related to people’s feelings, so many people begin to conduct emotional analysis and research on tweets. SemEval 2019 Task 5 is to detect hate speech on tweets. Task A is a binary classification task that predicts whether English or Spanish tweets for specific goals (women or immigrants) are hateful or not hateful (Basile et al., 2019). There are many studies that currently use tweets as a corpus for natural language processing (NLP). Text classification using traditional machine learning methods mainly includes Support Vector Machines (SVMs) (Gunn et al., 1998), Naive Bayes (McCallum et al., 1998) and Random Forests (Cutler et al., 2007), etc. In recent years, the use of deep neural networks for NLP has become mainstream, such as Convolutional Neural Networks (CNNs) for sentence classification (Kim, 2014) and Recurrent Neural Networks (RNNs) (Graves et al., 2013).

This task aims to predict whether the tweet for each ID is a hate speech about women or immigrants. Our system implements a stacked Bidirectional Gated Recurrent Units (BiGRUs) (Cho et al., 2014) based on a capsule network. The vector representations of words are obtained with fastTex. The result of the classification is through the output of a fully connected layer. The rest of this paper is organized as follows: Data Processing and analysis are discussed in section 2. Section 3 provides the details of the proposed model. Experiments and results are described in Section 4. Finally, we draw conclusions in Section 5.

2 Data Processing

This part describes the experimental data and data processing analysis of SemEval 2019 Task 5 subtask-A in English.

[*] Corresponding author

[1] http://twitter.com

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 535–539
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

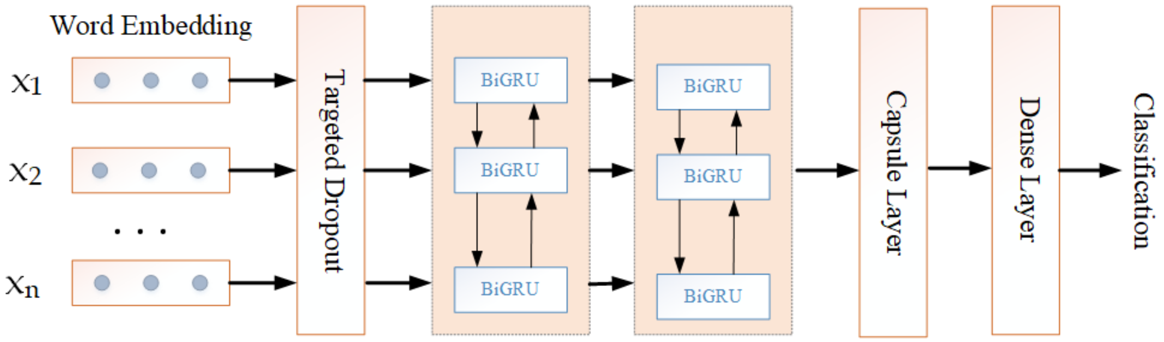

Figure 1: Neural architecture of stacked BiGRU with capsule network.

2.1 Experimental Data

This is a binary classification task of hate speech about immigrants or women. The task organizers provide training sets, development sets and test sets, respectively. Table 1 shows the data distribution of hate speech and non-hate speech in each data set. From Table 1, we can find that there are 9,000 tweets in training set, 1,000 tweets in development set and 2,971 tweets in test set.

data	hate speech	non-hate speech
training set	3,783	5,217
dev set	427	573
test set	1,252	1,719

Table 1: Distribution of labels in each datasets.

2.2 Processing Data

We perform a series of standard processing on datasets.

- All punctuation marks are removed.
- All characters are converted to lowercase.
- All hyperlinks are replaced by “url”.
- All sentences are tokenized by Natural Language Toolkit (NLTK) (Bird et al., 2009).
- All numbers are replaced by “number”
- All contractions are normalized, like place “shouldn’t” with “should not” and “dosen’t” with “does not” and so on.
- All @specific user names are replaced with usernames, for example “@PdxPatriot1” is replaced with “username”.

We consider the specific length of the sentence in the input model. If it is too long, the calculation time of the training model will increase. If it is too short, it will lose extra information. So we choose twice the average value, which is 45, as the final length of the sentence in the input model, so that the lost information will not be too much, and the calculation time will not be too long. In the training set, the development set and the test set have 473, 122, and 102 sentences respectively longer than 45, and the maximum sentence length is 65.

3 System Description

Our system can be roughly divided into two parts: the space vector representation of the words and the learning of the tweet content by the capsule network. We first map the words into a low-dimensional space vector, then feed the sentence vectors composed of these word vectors into a capsule network to learn the sentence features, and finally classify the text of the test set by a softmax function.

3.1 Word Representation

Representing a word by using a low-dimensional vector is currently the most common method in natural language processing. The fastText (Joulin et al., 2017) tool is used in our system to get the word representation of the sentences. A low-dimensional vector in fastText is associated with each word, and hidden representations can be shared between different classes of classifiers so that textual information can be used together in different classes. So fastText is a very efficient, word-based vectorization model for text classification. The pre-trained fastText embedding is used in our system[2].

3.2 Model Description

In order to enrich the word vector representation in the text, we use a stacked Bidirectional Gated Recurrent Units (BiGRUs) (Cho et al., 2014). The output of BiGRU is then used as the input to the capsule network (Sabour et al., 2017). The final result is obtained by the $softmax$ activation function in the fully connected layer. The model architecture is show in Figure 1.

Targeted Dropout Layer: *Dropout* regularization only activates some local neurons in each forward propagation, so it adds sparsity properties during training. This encourages the neural network to learn a representation that is robust to sparsification, that is, to randomly delete a set of neurons. *Targeted Dropout* (Gomez et al., 2018) sorts weights or neurons based on some measure of fast approximation weight importance and applies *Dropout* to those elements of lower importance. This approach encourages neural networks to learn more important weights or neurons. In other words, the network learns to be robust to our choice of post hoc pruning strategy. At the same time it is easy to implement with *Keras* [3].

Stacked BiGRU: To get more fine-grained sentence information, we use stacked Bidirectional Gated Recurrent Units (BiGRUs) to encode sentence information. The “stack” here refers to 2, which is 2 layers BiGRU. The information of the sentence is directional. The forward GRU can only get the information from the front to the back of the sentence, and can’t encode the information from the back to the front. BiGRU better captures semantic dependencies in both directions.

Capsule Layer: The capsule network (Sabour et al., 2017) replaces a single neuron node of a traditional neural network with a neuron vector, and trains a completely new neural network in the way of Dynamic Routing, which effectively improves the low efficiency and space insensitivity of the CNN model. The capsule network is connected the same way as a fully connected network. Each capsule neuron in the previous layer is connected to each capsule neuron in the next layer. Each connection of the capsule network is also weighted. The difference is that there is a coupling coefficient on the connection of the capsule network. The coupling coefficient is determined by the iterative dynamic routing process.

4 Experiments and Results

4.1 Evaluation

To evaluate the performance of the classification system, the system uses a standard evaluation metrics that includes *accuracy*, *precision*, *recall*, and F_1-*score*. In this task we use F_1-*score* to measure the performance of the proposed method. *Accuracy* is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. *Precision* is the ratio of correctly predicted positive observations to the total predicted positive observations. *Recall* is the ratio of correctly predicted positive observations to the all observations in actual class. F_1-*score* is the weighted average of *Precision* and *Recall*. *Precision* and *recall* have equal contributions to F_1-score. The formula for F_1-*score* is defined as:

$$F_1 - score = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \quad (1)$$

4.2 Hyperparameter

The *Targeted Dropout* layer has two parameters, *drop_rate* and *target_rate*. In this system, these two parameters are both set to 0.55.

For the stacked BiGRU, the first layer BiGRU *units* = 64, and the second layer BiGRU *units* = 64.

The parameters of the capsule layer are set as follows: *routings* = 5, the number of caspule is 10 and the dimension is 32.

Finally, at the full connection layer output, we added two parameters, *kernel_regularizer* and *activity_regularizer*, respectively. *Kernel_regularizer* uses l_2 regularization with a parameter of 0.001, *activity_regularizer* is l_1 regularization, and the parameter is also set to 0.001.

Usually the multi-classification problem uses *categorical crossentropy* as the loss function. But our system uses *binary crossentropy* in this binary classification.

We set *epochs* = 6 and *batch size* = 64.

4.3 Experiments and Result Analysis

We conduct several experiments to gain insight into the performance of the proposed model. First

[2] https://fasttext.cc/docs/en/english-vectors.html

[3] https://pypi.org/project/keras-targeted-dropout/

we compare the normal *Dropout* and *Targeted Dropout* performance.

It can be seen from Table 2 that the performance of *Targeted Dropout* is significantly better than that of *Dropout*. Model performance increases by 5% on average F_1*-score*.

Sets	Acc	P	R	F$_1$
Dropout	0.53	0.58	**0.61**	0.52
Targeted Dropout	**0.56**	**0.64**	0.60	**0.55**

Table 2: Experimental results on test set. The values in the table are macro averages.

To determine the specific parameters of the *Targeted Dropout*, we do a lot of comparison experiments. As can be seen from Table 3, the best parameter is 0.55. This is also the parameter we submitted to the system in the competition.

Targeted Dropout	Acc	P	R	F$_1$
0.40	0.53	0.58	0.63	0.50
0.45	0.56	0.60	0.63	0.54
0.50	0.53	0.59	**0.64**	0.49
0.55	**0.56**	**0.64**	0.60	**0.55**
0.60	0.55	0.60	0.63	0.54

Table 3: Experimental results of different Targeted Dropouts on the test set.

We compare the four network architectures based on a capsule network, LSTM, GRU, BiLSTM and BiGRU. We observe that the performance of BiGRU is better than the other three in this task. Compared to MFC baseline and SVC baseline, our method increases the average F_1*-score* by 0.18 and 0.10, respectively, as is shown in Table 4.

The values of MFC baseline and SVC baseline come from the data published by the organizer[4]. To ensure the fairness of the experiment, the parameters of the capsule network remain unchanged, using the parameters mentioned in section 4.2.

5 Conclusion and Future Work

In this paper, we present a stacked BiGRU model based on a capsule network system in the task "Shared Task on Multilingual Detection of Hate". We replace *Dropout* with *Targeted Dropout*, the effect is more obvious, indicating that *Targeted Dropout* is effective in this system. At the same time, we have conducted several experiments to find the optimal parameters of *Targeted Dropout*. Through comparative experiments, BiGRU is the best model based on capsule networks.

Model	Acc	P	R	F$_1$
MFC baseline	**0.58**	0.29	0.5	0.37
SVC baseline	0.49	0.60	0.55	0.45
LSTM	0.55	0.60	**0.64**	0.53
GRU	0.54	0.59	0.62	0.52
BiLSTM	0.53	0.58	0.62	0.51
BiGRU	0.56	**0.64**	0.60	**0.55**

Table 4: Each model is a stacked or two-layer model, and the units in the model are all 64.

Due to time limit, we don't tune the parameters of the capsule network. In the future, we will adjust the parameters of the capsule network to optimize the performance of the model. Secondly, we are going to try ensemble methods such as hard voting, soft voting and stacking to find the one that works best for our task. Finally, we would like to explore transfer learning technology.

Acknowledgments

This work was supported by the Natural Science Foundations of China under Grant Nos. 61463050, 61702443 and 61762091, and the Project of Innovative Research Team of Yunnan Province under Grant No. 2018HC019.

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Rangel, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*,

[4] https://docs.google.com/spreadsheets/d/1wSFKh1hvwwQIoY8_XBVkhjxacDmwXFpkshYzLx4bw-0/edit#gid=0

pages 1724–1734. Association for Computational Linguistics.

D Richard Cutler, Thomas C Edwards Jr, Karen H Beard, Adele Cutler, Kyle T Hess, Jacob Gibson, and Joshua J Lawler. 2007. Random Forests for Classification in Ecology. *Ecology*, pages 2783–2792.

Aidan N Gomez, Ivan Zhang, Kevin Swersky, Yarin Gal, and Geoffrey E Hinton. 2018. Targeted Dropout. In *International Conference on Neural Information Processing Systems*.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech Recognition with Deep Recurrent Neural Networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Steve R Gunn et al. 1998. Support Vector Machines for Classification and Regression. *ISIS technical report*, 14(1):5–16.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistic-s.

Andrew McCallum, Kamal Nigam, et al. 1998. A Comparison of Event Models for Naive Bayes Text Classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

John T Nockleby. 2000. Hate Speech. *Encyclopedia of the American constitution*, 3(2):1277–1279.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, volume 10, pages 1320–1326.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic Routing Between Capsules. In *Advances in neural information processing systems*, pages 3856–3866.