DMCB at SemEval-2019 Task 5: LSTM with Self Attention model for Hate Speech Detection

Hajung Sohn, Hyunju Lee
Gwangju Institute of Science and Technology,
Data mining and Computational Biology Lab, Gwangju, Korea
{hajungsohn, hyunjulee@gist.ac.kr}

Abstract

In this paper we describe a neural architecture developed for SemEval-2019 Task 5 HatEval. We used a recurrent neural network(LSTM) with self attention to learn the latent context of tweets. Static GloVe word embeddings and Part-of-Speech(POS) embeddings were also used to effectively capture semantic and syntactic features. The GloVe embeddings were pre-trained in a large twitter corpus that consists of 2 billion tweets, and POS information was tagged by the SpaCy library. Morover, slur tags were inserted before and after slur words to utilize lexicons as prior information. Our model shows that simply using POS information, slur tags with recurrent neural networks outperforms several baselines. At the competition, we achieved an accuracy of 71% and average f1 score of 70% in the Spanish test dataset.

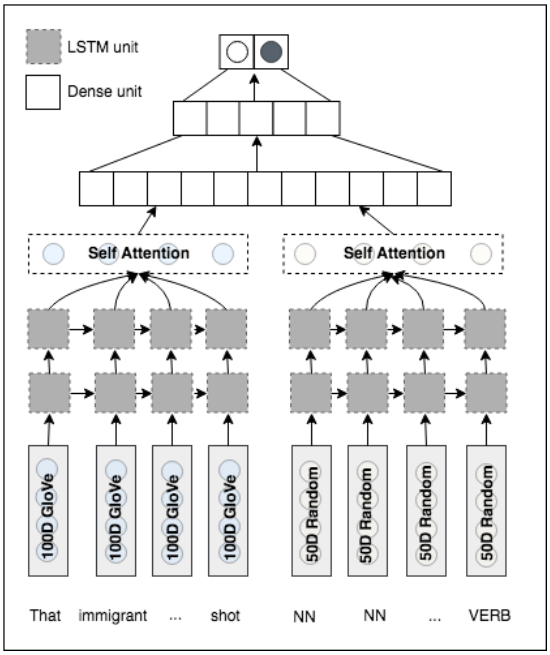
1 Introduction

Automatic Hate Speech Detection is a task with growing importance due to the rapid growth of social media such as Twitter, Facebook and community forums. Social medias changed ways of communication and content publishing, but they are also increasingly exploited by hate speech. This paper introduces a Deep Neural Network that participated for the SemEval-2019 Task 5 subtask A and B (Basile et al., 2019). Task A is to automatically detect hate speech in tweets towards immigrants and women and Task B is additionally detecting whether the incitement is against an individual rather than against a group of people, and whether an aggressive behavior of the author can be identified as a prominent feature of the expression of hate. The traditional approach for automatic hate speech recognition is to use lexical features such as making a dictionary of insulting keyword or phrases and filter out hate speech based on this dictionary. However, this method neglects hate speeches without any words in the insulting and abusing language dictionary. Recently, neural network approaches have outperformed existing methods for many text classification problems, so a deep learning model can be introduced in this task, too. Hate speech recognition is similar to sentiment analysis and a sentence with negative polarity can generally be considered to correspond with a hate speech message. Our model used methods applied in sentiment analysis such as word embedding, recurrent neural network, attention in this task. Moreover, we used lexicon features and POS information as prior knowledge to effectively perceive the semantic and syntactic feature of the tweet.

2 Related Works

Previous rule based methods make use of black list dictionary and regular expressions. However, these methods fall short when contending with more subtle, less ham-fisted examples of hate speech. The dictionary based classification will have a high recall but also a high false positive rate because it tends to classify sentences that has swear words as hate speech(Burnap and Williams, 2015). Several machine learning methods were introduced to overcome the downsides of dictionary based methods. Regression was used to classify hate speech in comments. The feature used in comments were n-grams, linguistic, syntactic, and distributional semantics. Linguistic features include the length of each comment, the average length of word, the number of punctuations, the number of politeness words. Syntactic features include POS tags and dependency relations(Nobata et al., 2016). SVM was also used to classify tweets to 3 categories, hate speech, offensive language , and neither. In this paper, the hate speech con-

Figure 1: The basic LSTM-Attention model used in this task.

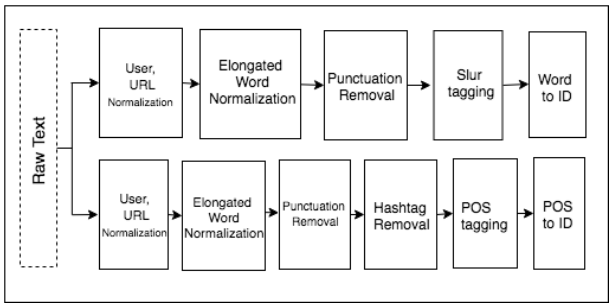


ditions were very strict and only took 5% of the total labeled dataset. Their works emphasized the importance of subtle linguistic meanings inside the context(Davidson et al., 2017). Deep neural networks were used in more recent researches.A CNN-LSTM model was proposed to classify tweets to categories of racism, sexism, and neither of them. The CNN layer was first used to capture meaningful tokens in tweets and the following LSTM layer was used to find the dependency between important features(Zhang and Luo, 2018). A method using LSTM model with random word embeddings was also introduced to classify racist, sexist, and neutral tweets(Pitsilis et al., 2018).

3 Model

For this task,the Long Short Term Memory(LSTM) was used as a recurrent neural network to extract the ordering and semantic features of the sentence input. Self Attention was used to learn the importance of each tokens. Other methods, word embeddings, self attention, and lexicon based methods were also applied to improve performance. The overall model is depicted in Figure 1.

Figure 2: The Text and POS preprocessing pipeline.



3.1 Text preprocessing

We applied normalization on tweets using the ekphrasis library[1], which is a tweet specialized text preprocessing library(Baziotis et al., 2017). The url in each tweet was replaced by the <url> tag and the user name was replaced by the <user> tag. Then, elongated words such as 'yaaaaay' was normalized to be 'yaay'. While hashtags and emojis were not removed and used to be embedded, they were removed for text for Part of Speech(POS) tagging. Punctuations and special characters were removed to make the text cleaner. Then, each tweet was tokenized based on spaces. The preprocessing process is depicted in Figure 2

3.2 Word Embeddings

Word Embedding is a way to represent words as vectors. Words are usually embedded to vectors by its distributional characteristics. In this paper, word vectors that were pre-trained on a large twitter corpus by GloVe(Pennington et al., 2014) were used to encode work tokens into vectors. Well trained word vectors have high cosine similarity if they have a similar meaning or occur in a similar context. For example, 'apple' and 'banana' should have a higher cosine similarity than 'apple' and 'steal'. For this characteristic, word vectors are generally used as a way to encoding words to numbers that are fed to the neural network rather than using Bag-of-Words.

3.3 Long Short Term Memory

Long Short Term Memory is a recurrent neural network that was introduced to overcome the shortcomings of Recurrent Neural Network. The Long term dependency problem is relaxed in the

[1]https://github.com/cbaziotis/ekphrasis

LSTM by introducing a forget gate that learns to discard information(Hochreiter and Schmidhuber, 1997).

3.4 Attention

Attention mechanism was first introduced in neural translation and showed an improvement in performances(Bahdanau et al., 2014). Sentence classification can also use attention to find more important features that affect the classification result. Attention helps the network to learn the importance of each node by calculating the attention score. The attention mechanism assigns a weight a_i to each word annotation h_i. Then the attention score is multiplied to get the overall representation r. This self attention mechanism was suggested by (Raffel and Ellis, 2015) and (Rocktäschel et al., 2015).

$$e_i = tanh(W_h h_i + b_h), e_i \in [1, 1] \quad (1)$$

$$\frac{exp(e_i)}{\sum_{t=1}^{T} exp(e_t)} = 1, \sum_{i=1}^{T} a_i = 1 \quad (2)$$

$$r = \sum_{i=1}^{T} a_i h_i \quad (3)$$

3.5 Part of Speech(POS) Tagging

The Part of Speech information of each tweet was used to utilize the syntactic information. After tweet data were cleaned, they were tagged with Part of Speech(POS) labels by the SpaCy library.[2]

3.6 Lexicon based method

A list of offensive language was used to detect slur words and a <slur>, </slur> tag was placed before and after each word. The list of slur words were downloaded from CMU research page. [3]

4 Experiments and Results

4.1 Dataset

The English dataset consists of 9000 tweets as training data, and 1000 tweets as validation data, 2971 tweets as test data. Each sentence is labeled by (a)whether it is a hate speech toward immigrants or women, (b)whether it is targeting an individual or a group, and (c) whether it infers a more active behavior or not. Task A is to develop a model that well classifies (a). Task B is to first classify (a) and then further predict (b) and (c). In this paper, only experiments on Task A is done. The statistics of train data and validation data is shown in Table 1.

Table 1: Number of examples for each class in Train, Dev, and Test Dataset

	Train	Validation	Test
not hate	5217(58%)	573(57%)	1721(58%)
hate	3783(42%)	427(43%)	1250(42%)
total	9000	1000	2971

4.2 Experimental Setup

When training the model, the batch size was 50 and . The optimizer used was Adam with the learning rate was 0.001. Regularization and Dropout were also used to prevent overfitting and make the model more robust to noise. The hyperparameters used for the neural network is summarized in Table 2.

Table 2: Hyper-Parameters

Batch size	50
Learning Rate	0.001
Optimizer	Adam
Weight decay	0.00001
Dropout	0.3
Recurrent Dropout	0.3
Loss function	categorical cross entropy

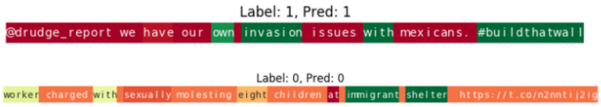
4.3 Results and Discussion

The results of the experiment is stated in Table 3. Since convolutional Neural Network(CNN) is a widely used in text classification, (Kim, 2014) CNN along with Long short term memory(LSTM) with different layer numbers were experimented too. We used pretrained twitter word embeddings with 100 dimension as an input for each model. Attention layer and ¡slur¿ tags were added to the LSTM layer for better performance. POS embeddings were also used to extract syntatic information. LSTM model was better in performance than CNN, which had an 74% accuracy. LSTM with 2

[2]https://spacy.io/

[3]https://www.cs.cmu.edu/ biglou/resources/

Figure 3: Visualization of the Attention score of tweets. The green colored words has a higher attention score than yellow and red colored words.



layers was better than a single layer, which scored 75% accuracy. A 2 layer LSTM with Self Attention and SlurTag with POS embedding showed the best performance which is 78%. Large difference between development and test accuracy seems to be due to difference in domains. While development data had equal cases of immigrant and women hate speech, the test set has a higher proportion of women related hate speech.

Table 3: English Development and Test set Accuracy

Model	Dev	Test
SVC	-	49%
CNN + GloVe	74%	51%
1 layer LSTM + GloVe	75%	51%
2 layer LSTM + GloVe	76%	51%
2 layer LSTM + GloVe + Attention	77%	52%
2 layer LSTM + GloVe + Attention + SlurTag	78%	52%
2 layer LSTM + GloVe + Attention + SlurTag + POS	78%	53%

4.4 Attention Visualizations

The visualization of attention scores is depicted in figure 3. The green colored words has a higher attention score than yellow and red colored words. In the first hate tweet, words like 'invasion' and 'buildthatwall' which is more related to hate speech had a higher attention score than than other words. For s non hate speech example, words like 'immigrant' and 'shelter' which is less relevant to hate speech has higher attentions scores.

5 Conclusion

This paper introduced an LSTM and self attention based Neural Network for automatic hate detection on tweets and describes analysis on several experiments. We used static GloVe word embeddings and POS embeddings to both utilize semantic and syntactic information. Moreover, we used slur word lexicons to employ prior information useful in hate speech recognition.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

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, location = Minneapolis, Minnesota.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

P. Burnap and M. L. Williams. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. In *Policy Internet 7*, pages 223–242.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *CoRR*, abs/1703.04009.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 145–153, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Georgios Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Detecting offensive language in tweets using deep learning.

Colin Raffel and Daniel P. W. Ellis. 2015. Feed-forward networks with attention can solve some long-term memory problems. *CoRR*, abs/1512.08756.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *CoRR*, abs/1509.06664.

Ziqi Zhang and Lei Luo. 2018. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *CoRR*, abs/1803.03662.