KDEHatEval at SemEval-2019 Task 5: A Neural Network Model for Detecting Hate Speech in Twitter

Umme Aymun Siddiqua, Abu Nowshed Chy, and Masaki Aono
Department of Computer Science and Engineering
Toyohashi University of Technology, Toyohashi, Aichi, Japan.
{aymun,nowshed}@kde.cs.tut.ac.jp and aono@tut.jp

Abstract

In the age of emerging volume of microblog platforms, especially twitter, hate speech propagation is now of great concern. However, due to the brevity of tweets and informal user generated contents, detecting and analyzing hate speech on twitter is a formidable task. In this paper, we present our approach for detecting hate speech in tweets defined in the SemEval-2019 Task 5. Our team KDEHatEval employs different neural network models including multi-kernel convolution (MKC), nested LSTMs (NLSTMs), and multi-layer perceptron (MLP) in a unified architecture. Moreover, we utilize the state-of-the-art pre-trained sentence embedding models including DeepMoji, InferSent, and BERT for effective tweet representation. We analyze the performance of our method and demonstrate the contribution of each component of our architecture.

1 Introduction

Nowadays, microblog platforms such as twitter has become the most popular communication medium among the people due to its convenient feature for sharing views, opinions, breaking news, and ideas. Besides its robust communication feature, it has facilitated the evil-minded people to propagate anti-social behavior including online harassment, cyber-bullying, and hate speech.

Hate speech is commonly defined as any communication that disparages a person or a group on the basis of some characteristics such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics (Basile et al., 2019). Given the massive amount of user-generated contents on the microblog, the problem of detecting, and therefore possibly limit the hate speech diffusion, is becoming fundamental, for instance for fighting against misogyny and xenophobia.

To address the challenges of hate speech detection in microblog platforms, Basile et al. (2019) proposed a multilingual detection of hate speech (HatEval) in twitter, task 5 at SemEval-2019. The task features two specific different targets including immigrants and women and focuses on two related subtasks.

Task A defines a two-class (or binary) classification problem where a system needs to predict whether a tweet in English or Spanish with a given target (women or immigrants) is hateful or not hateful. Whereas task B defines the aggressive behavior and target classification problem. A system first classifies a hateful tweet as aggressive or not aggressive, and then identify the target harassed as the individual or generic (i.e., single human or group). In this paper, we only focus on the English tweets for both task A and B.

The rest of the paper is structured as follows: **Section 2** provides a brief overview of prior research. In **Section 3**, we introduce our proposed neural network model. **Section 4** includes experiments and evaluations as well as the analysis of our proposed method. Some concluded remarks and future directions of our work are described in **Section 5**.

2 Related Work

Early studies on hate speech detection focused mainly on lexicon-based approaches (Kwok and Wang, 2013; Gitari et al., 2015). However, these approaches prone to failure for detecting hate speech in a microblogging platform where rare terms are evolving incessantly. Besides some researchers tackled the problem by employing feature (e.g., N-gram, TF-IDF) based supervised learning approach using SVM and Naive-Bayes classifier (Gaydhani et al., 2018; Unsvåg and Gambäck, 2018).

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 365–370
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

More recently several researchers tried to address the problem by using state-of-the-art neural network based models. Among several prominent works, Badjatiya et al. (2017) employed multiple deep learning architectures including CNNs, LSTMs, and fastText to learn semantic word embeddings for hate speech detection. Golem et al. (2018) utilized the combination of traditional shallow machine learning models and deep learning models for hate speech detection. Pitsilis et al. (2018) utilized the ensemble of recurrent neural network (RNN) classifiers and incorporated various features associated with user-related information. Zhang et al. (2018) introduced a new method by combining a convolutional neural network (CNN) and gated recurrent unit (GRU) models. Djuric et al. (2015) used the comment embeddings for detecting hate speech.

3 Proposed Framework

In this section, we describe the details of our proposed neural network model. The goal of our proposed approach is to detect whether a tweet is hateful or not as well as determine its aggressiveness and identify the target harassed as individual or generic. We consider each task as a binary classification problem as well as train and evaluate our model accordingly. Figure 1 depicts an overview of our proposed model.

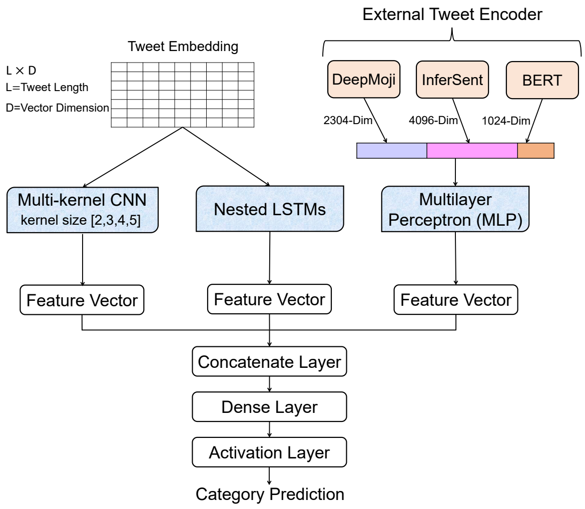

Figure 1: Proposed framework.

At first, we utilize a pre-trained word embedding model to obtain the high-quality distributed vector representations of tweets. Next, we apply the multi-kernel convolution (MKC) and nested LSTMs (NLSTMs) models to extract the higher-level feature sequences with sequential information from the tweet embeddings. Besides, we employ three different pre-trained tweet encoder models including DeepMoji, InferSent, and BERT to encode each tweet into 2304, 4096, and 1024-dimensional feature vector, respectively. These feature vectors are then combined and sent to a multi-layer perceptron (MLP) module. Finally, the generated output feature sequences from MKC, NLSTMs, and MLP are concatenated and fed into the fully-connected prediction module to determine the final category label. For the simplicity of discussion, we named our proposed neural network architecture as MKC-NLSTMs-MLP. Next, we describe each component elaborately.

3.1 Embedding Layer

Distributed representation of words known as word embedding is treated as one of the most popular representations of documents vocabulary due to its capability of capturing the context of a word within a document as well as estimating the semantic similarity and relation with other words (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017).

In our proposed framework, we utilize a pre-trained word embedding model based on fastText (Bojanowski et al., 2017) to obtain the high-quality distributed vector representations of tweets. The dimensionality of the embedding matrix will be $L \times D$, where L is the tweet length, and D is the word-vector dimension.

3.2 Multi-kernel Convolution

The convolution layers usually applied to extract the higher level features from the given input matrix. Since kernel sizes, i.e., the size of the convolution filters have a significant effect on performance, we apply filters with different sizes to get the different kinds of effective features. In our multi-kernel convolution, We perform the convolution on the input tweet's embedding matrix by using four different kernel sizes: 2, 3, 4, and 5. Some previous studies already demonstrated the effectiveness of multi-kernel convolution over the single one (Kim, 2014; Zhang and Wallace, 2015; Wang et al., 2017).

3.3 Nested LSTMs

In nested LSTMs (NLSTMs) (Moniz and Krueger, 2017), the LSTM memory cells have access to their inner memory, where they can selectively read and write relevant long-term information. While the value of the outer memory cell in the LSTM is estimated as $c_t^{outer} = f_t \odot c_{t-1} + i_t \odot g_t$, memory cells of the NLSTM use the concatenation $(f_t \odot c_{t-1}, i_t \odot g_t)$ as input to an inner LSTM (or NLSTM) memory cell, and set $c_t^{outer} = h_t^{inner}$. The inner memories of NLSTMs operate on longer time-scales and capture the context information from the input tweets effectively.

3.4 Pre-trained Models for Feature Encoding

In order to extract features for effective tweet representation, we utilize the three state-of-the-art pre-trained sentence embedding model including DeepMoji, BERT, and InferSent. In this section, we briefly describe these models.

DeepMoji: DeepMoji (Felbo et al., 2017) performs distant supervision on a dataset of 1246 million tweets comprising a more diverse set of noisy labels. DeepMoji uses an embedding layer of 256 dimensions to project each word of a tweet into a vector space. Two bidirectional LSTM layers with 1024 hidden units in each (512 in each direction) are applied to capture the context of each word. Finally, an attention layer takes all of these layers as input using skip-connections. We employ the representation vector of dimension 2304 obtained from the attention layer as the features.

BERT: BERT (Devlin et al., 2018) stands for Bidirectional Encoder Representations from Transformers, which is a new method of pre-training sentence representations. We employ the BERT-Large, Uncased model to encode each tweet into a 1024-dimensional feature vector.

InferSent: InferSent (Conneau et al., 2017) is a universal sentence embedding model trained using the supervised data of the Stanford Natural Language Inference (SNLI) datasets. We employ the InferSent model trained on fastText (Bojanowski et al., 2017) vectors to encode each tweet into a 4096-dimensional feature vector.

3.5 Multi-layer Perceptron

After extracting features from the pre-trained external tweet encoder model, we concatenate them and pass to a fully connected multi-layer perceptron (MLP) network.

A multilayer perceptron (MLP) (Pedregosa et al., 2011) is a feed-forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron with a nonlinear activation function. MLP utilizes a supervised learning technique called back-propagation for training the network.

3.6 Prediction Module and Model Training

We concatenate the final tweet representation from the multi-kernel convolution (MKC) module, NLSTMs module, and MLP module and pass it to a fully connected softmax layer for category prediction. We consider cross-entropy as the loss function and train the model by minimizing the error, which is defined as:

$$E(x^{(i)}, y^{(i)}) = \sum_{j=1}^{k} 1\{y^{(i)} = j\} \log(y_j^{\sim(i)})$$

where $x^{(i)}$ is the training sample with its true label $y^{(i)}$. $y_j^{\sim(i)}$ is the estimated probability in $[0, 1]$ for each label j. $1\{condition\}$ is an indicator which is 1 if true and 0 otherwise. We use the stochastic gradient descent (SGD) to learn the model parameter and adopt the Adam optimizer (Kingma and Ba, 2014).

4 Experiments and Evaluations

4.1 Dataset Collection

The multilingual detection of hate speech (HatEval) task 5 at SemEval-2019 (Basile et al., 2019) provides a benchmark dataset to evaluate the performance of the participants' systems. The proposed task features two specific different targets including immigrants and women in a multilingual perspective, for Spanish and English. However, we only used the English dataset to evaluate our proposed system. The training, validation, and test set of the English dataset contains the 9000, 1000, and 2971 annotated tweets, respectively.

4.2 Model Configuration

In the following, we describe the set of parameters that we have used to design our proposed neural network model, MKC-NLSTMs-MLP. Our designed model was based on Tensorflow (Abadi et al., 2016) and trained on a GPU (Owens et al., 2008) to capture the benefit from the efficiency of parallel computation of tensors. We performed hyper-parameter optimization using a simple grid search. We used the 300-dimensional fastText embedding model pre-trained on Wikipedia with skip-gram (Bojanowski et al., 2017) to initialize the word embeddings in the embedding layer described in Section 3.1. For the multi-kernel convolution described in Section 3.2, we employed 4 different kernel sizes including (2,3,4,5), and the number of filters was set to 600. The nested LSTMs module contains 2 layers and multilayer perceptron (MLP) module contains 3 fully-connected dense layers. We trained all models with 15 epochs with a batch size of 32 and an initial learning rate 0.001 by Adam optimizer. The MLP layers were dropped out with a probability of 0.02. $L2$ regularization with a factor of 0.01 was applied to the weights in the softmax layer. Unless otherwise stated, default settings were used for the other parameters.

4.3 Evaluation Measures

To evaluate the performance of the system, the organizers used different strategies and metrics for the task A and B (Basile et al., 2019). For the task A, standard evaluation metrics, including accuracy, precision, recall, and F1-score were applied to estimate the performance of a system. However, F1-score considered as the primary evaluation measure for this task.

For the task B, macro average $F1$-score of the hate speech (HS), target range (TR), and aggressiveness (AG) category and exact match ratio (EMR) of these categories are used as the evaluation measures. EMR considered as the primary evaluation measure for task B.

4.4 Experimental Results

We now evaluate the performance of our proposed method, MKC-NLSTMs-MLP, in this section. The summarized results for task A and task B are presented in Table 1 and Table 2, respectively.

At first, we presented the performance of our proposed method denoted by team name KDEHatEval as well as presenting the performance of randomly chosen top-ranked participated systems and HatEval-2019 baselines. The organizers used the SVC (a linear support vector machine) and MFC (a trivial model that assigns the most frequent label, estimated on the training set) as the baseline system (Basile et al., 2019).

Team Name	Accuracy	Precision	Recall	F1-Score
KDEHatEval	0.493	0.633	0.555	0.440
Fermi	0.653	0.690	0.679	0.651
YNU_DYX	0.560	0.636	0.603	0.546
SINAI_DL	0.535	0.601	0.577	0.519
Hateminers	0.544	0.658	0.596	0.516
SVC Baseline	0.492	0.595	0.549	0.451
MFC Baseline	0.579	0.289	0.500	0.367

Table 1: (Task A) Our result with other selected teams.

Team Name	Avg. F1-Score	Exact Match Ratio (EMR)
KDEHatEval	0.559	0.324
MFC Baseline	0.421	0.580
CIC-1	0.551	0.568
SINAI_DL	0.611	0.384
Hateminers	0.589	0.357
SVC Baseline	0.578	0.308

Table 2: (Task B) Our result with other selected teams.

In order to estimate the effect of each component of our MKC-NLSTMs-MLP model, we performed the component ablation study. In this regard, we removed one component each time and repeated the experiment. The summarized experimental results of component ablation study for the task A are presented in Table 3.

Method	Accuracy	Precision	Recall	F1-Score
MKC-NLSTMs-MLP	0.493	0.633	0.555	0.440
−MKC	0.441	0.507	0.503	0.381
−NLSTMs	0.483	0.630	0.548	0.423
−MLP	0.495	0.636	0.557	0.443
−MKC−NLSTMs	0.458	0.507	0.505	0.431
−MKC−MLP	0.475	0.592	0.537	0.419
−NLSTMs−MLP	0.485	0.640	0.549	0.423

Table 3: (Task A) Ablation study of our proposed method.

From the results, it can be observed that when removing multi-kernel convolution (MKC) and nested LSTMs (NLSTMs) the overall F1 score decreased by 5.6% and 1.7%, respectively. However, when removing external tweet embedding with MLP module, the results increased by 0.3%. This observation deduced that in our current architecture, external pre-trained model features with MLP contributed negatively.

Besides, removing one component, we also perform ablation study by removing two components and present the results accordingly in Table 3. This analysis provides the overall performance of the individual component.

5 Conclusion

In this paper, we presented our approach to the SemEval-2019 Task 5: HatEval: Detection of hate speech. We tackled the problem by employing several deep learning techniques including multi-kernel convolution, nested LSTMs, and multi-layer perceptron in a unified architecture.

Though we have used the state-of-the-art techniques in our proposed approach, the overall performance is not satisfactory. The contribution of nested LSTMs (NLSTMs) is not significant while compared with the multi-kernel convolution (MKC). Regarding this, one possible solution will be used the MKC on top of NLSTMs. Moreover, we observed that the multi-layer perceptron (MLP) model trained with the concatenated features from the pre-trained sentence embedding models hurts the performance of our proposed architecture. We need to observe the ablation study of the sentence embedding models as well as modify the MLP architecture to mitigate this issue.

Therefore, there is much room left to improve the performance of our method presented in HatEval-2019 task. In the future, we have a plan to overcome these limitations by introducing several sophisticated techniques.

Acknowledgments

The part of this research is supported by MEXT KAKENHI, Grant-in-Aid for Scientific Research (B), Grant Number 17H01746 and by grants from the KDDI Foundation.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI)*, pages 265–283. USENIX Association.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web (WWW) Companion*, pages 759–760. International World Wide Web Conferences Steering Committee.

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.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics (TACL)*, 5:135–146.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680, Copenhagen, Denmark. ACL.

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*.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*, pages 29–30. ACM.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1615–1625. ACL.

Aditya Gaydhani, Vikrant Doma, Shrikant Kendre, and Laxmi Bhagwat. 2018. Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. *arXiv preprint arXiv:1809.08651*.

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering (IJMUE)*, 10(4):215–230.

Viktor Golem, Mladen Karan, and Jan Šnajder. 2018. Combining shallow and deep learning for aggressive text detection. In *Proceedings of the 1st Workshop on Trolling, Aggression, and Cyberbullying (TRAC-2018)*, pages 188–198.

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. ACL.

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

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: detecting tweets against blacks. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, pages 1621–1622. AAAI Press.

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 (NeurIPS)*, pages 3111–3119.

Joel Ruben Antony Moniz and David Krueger. 2017. Nested lstms. In *Proceedings of the 9th Asian Conference on Machine Learning (ACML)*, pages 530–544. Springer.

John D Owens, Mike Houston, David Luebke, Simon Green, John E Stone, and James C Phillips. 2008. Gpu computing. *Proceedings of the IEEE*, 96(5):879–899.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research (JMLR)*, 12:2825–2830.

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

Georgios K Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Detecting offensive language in tweets using deep learning. *arXiv preprint arXiv:1801.04433*.

Elise Fehn Unsvåg and Björn Gambäck. 2018. The effects of user features on twitter hate speech detection. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 75–85.

Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. 2017. Combining knowledge with deep convolutional neural networks for short text classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2915–2921. AAAI Press.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *Proceedings of the 15th European Semantic Web Conference (ESWC)*, pages 745–760. Springer.