YNU_wb at HASOC 2019: Ordered Neurons LSTM with Attention for Identifying Hate Speech and Offensive Language

Bin Wang, Yunxia Ding, Shengyan Liu, and Xiaobing Zhou(✉)

School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
zhouxb@ynu.edu.cn

Abstract. The paper describes the system submitted to HASOC2019: Hate Speech and Offensive Content Identification in Indo-European Languages. The task aims to categorize offensive language in social media, we only participated in Sub-task A for English, which aims to identify offensive language and hate speech. In order to address this task, we proposed a system based on an ordered neurons LSTM with an attention model, and used a K-folding approach to ensemble. Our model achieved the Macro F1-score of 0.7882 and the Weighted F1-score of 0.8395 in the Subtask A for English language, and achieved the highest result.

Keywords: Hate speech · offensive language · ordered neurons LSTM · Attention.

1 Introduction

With the popularization of the Internet, more and more people are communicating on online social platforms. Therefore, the hate speech and offensive language in the social network have been paid increasing attention by people[1]. Hate speech and offensive language in online socialization have seriously affected people's daily life, such behavior could even lead to depression or suicide. Many social media companies and technology companies are currently researching the recognition of hate speech and offensive language. So whether the system can effectively identify hate speech and offensive language is a big challenge[9].

HASOC2019 is proposed for identifying hate speech and offensive content in Indo-European Languages [2]. Its purpose is to develop powerful technologies that can cope with multilingual data, and to develop a transfer learning method that can utilize the cross-language data. The task has three subtasks and three languages(English, code-mixed Hindi and German), in which Subtask A is a coarse-grained binary classification that the participating systems need

to classify tweets into two categories: hate and offensive (HOF) and non-hate and offense (NOT), Subtask B is a fine-grained classification that is reclassified on the hate speech and offensive language that subtask A has distinguished and Subtask C is to distinguish whether the target is an individual or a group.

In this competition, we only participated in Subtask A for English language. For Subtask A, we used a deep learning method to build an ordered neurons LSTM(ON-LSTM) with an attention model. ON-LSTM differs from the original LSTM in the way that it encodes the hierarchical structure of sentences into features to enhance the expressive power of LSTM[11]. Our model used ordered neurons LSTM to encoding sentences and used the attention mechanism to give each word in the sentence a different weight. In the training process, we used the OLID dataset[13] as the training dataset to train the model of this task. Finally, we used the K-folding method to ensemble, and got the highest result.

The structure of this paper is as follows: In section 2 we introduce some related work on identifying hate speech and offensive language. In section 3, we describe the datasets and how to build the model. In section 4 we describe the experimental results and analysis.

2 Related Work

In recent years, the topic of identifying hate speech and offensive language has attracted the attention of a large number of researchers in industry and academia. Fortuna and Nunes believed that the field of automatic detection of hate speech in the text is very important for online social platforms and has unquestionable social impact potential[4]. In this section, we will review some of the studies and briefly discuss their findings.

Gemeval2018[12] is about the identification of offensive language and the purpose of this task is to promote the study of offensive content recognition in German language microposts. The optimal system used five disjoint sets of features sets to train three basic classifiers (maximum entropy and two random forests ensembles), and then used a maximum entropy meta-level classifier for final classification[10]. HatEval is about the multilingual detection of hate speech against immigrants and women in Twitter[3]. The FERMI team as the best team for hatEval, proposed a SVM model with RBF kernel, exploiting sentence embeddings from Google's Universal Sentence Encoder as features[5]. OffensEval[14] is about the identifying and categorizing offensive language in social media. NULI team as the best performers used BERT-base-uncased with default-parameters[6].

As can be seen from the above, most of the methods for obtaining optimal results are machine learning models, and the deep learning models have not achieved good results. However, in this task, the deep learning model we used obtained the best results.

3 Methodology and Data

3.1 Data description

In this task, we used an extra dataset OLID[13], which is proposed by OffensEval mainly from Twitter. In Sub-task A, the purpose is to distinguish whether the tweet is hate and offensive (HOF) and non- hate and offensive (NOT). In which NOT: the text does not contain any hate speech, offensive content. HOF: the text contains Hate, offensive, and profane content [2]. For English language, The training dataset has a total of 5852 data, of which there are 3591 of NOT and 2261 of HOF, the ratio is about 3:2, the data is slightly unbalanced, for the OLID dataset, there are 8840 of NOT and 4400 of OFF.

3.2 ON-LSTM with an attention model

Our network architecture is shown in Figure 1. Our model is built on ON-LSTM[11] with attention, where the ON-LSTM is a variant of LSTM. Next we will explain the details of our system.

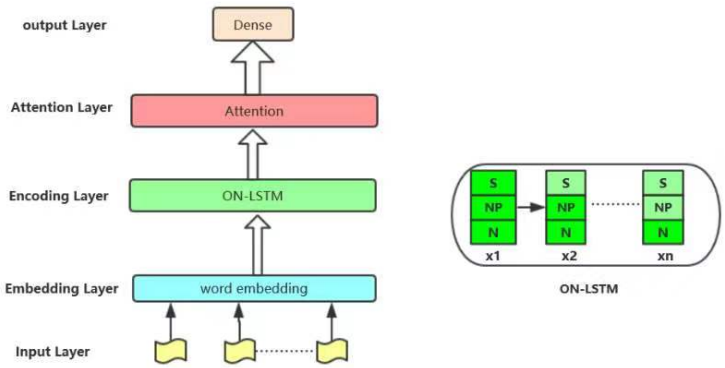

Fig. 1. The architecture of ON-LSTM with an attention model for Subtask A

- **Input layer:** This layer inputs all preprocessed text into the model.
- **Embedding layer:** This layer vectorizes the words in the existing dictionary that are entered by the pre-trained word vector model.
- **Encoding layer:** In this layer, we encode vectorized text with ON-LSTM, which sorts the neurons in a specific order, allowing the hierarchical structure (tree structure) to be integrated into the LSTM to express richer information. The gate structure and output structure of ON-LSTM are still similar to the original LSTM. The difference is that the update mechanism from $\widehat{c}_t$ to c_t is different. The formula is as follows[11]:

$$\widetilde{f}_t = \overrightarrow{cs}(softmax(W_{\widetilde{f}}x_t + U_{\widetilde{f}}h_{t-1} + b_{\widetilde{f}})) \tag{1}$$

$$\widetilde{i}_t = \overleftarrow{cs}(softmax(W_{\widetilde{i}}x_t + U_{\widetilde{i}}h_{t-1} + b_{\widetilde{i}})) \tag{2}$$

$$\omega_t = \widetilde{f}_t \circ \widetilde{i}_t \tag{3}$$

$$c_t = \omega_t \circ (f_t \circ c_{t-1} + i_t \circ \widehat{c}_t) + (\widetilde{f}_t - \omega_t) \circ c_{t-1} + (\widetilde{f}_t - \omega_t) \circ c_t \tag{4}$$

where $\overleftarrow{cs}$ and $\overrightarrow{cs}$ represents the explain briefly to the left and right respectively, the $\widetilde{f}_t$ and $\widetilde{i}_t$ are called master forget gate and master input gate respectively. ω_t gives the vector where the intersection is 1 and the rest is 0. In this way, the high-level information may be stored for a long distance, and the low-level information may be updated at each step of input, so the hierarchical structure is embedded by information hierarchy.

- **Attention layer:** The main function of this layer is to assign a weight to each word in the sentence, making the words that are biased towards hate speech and offensive language more prominent, in order to classify the sentence. Because all words in a sentence can be different in their expression of emotions[8]. Some emotional words can greatly influence whether a sentence is hate speech and help identify the hate category.
  First, we feed the word annotation h_i, and through a nonlinear layer to get a deeper representation u_i. Then, we calculate the similarity between u_i and the word-level context vector u_s and use the softmax function to get the normalized weight α_i. Finally, we calculate the sentence vector s by the weighted sum of the word annotation h_i. The specific formula is as follows:

$$u_i = \tanh(W_s * h_i + b_s) \tag{5}$$

$$\alpha_i = \frac{exp(u_i^T * u_s)}{\sum_i exp(u_i^T * u_s)} \tag{6}$$

$$s = \sum_i \alpha_i * h_i \tag{7}$$

- **Output layer:** This layer classifies and predicts the final aggregated information. This layer consists of only one Dense layer with sigmoid activation function.

3.3 K-folding ensemble

In this paper, we use a K-fold ensemble approach to enhance the performance of the model. The idea of this method is to cross-validate the source and K-fold. We randomly divide the training set into K parts and use the $K-1$ subsets to do the training. The remaining subset is used as the verification set, and K times are repeated. Finally, the K results are subjected to an accumulation averaging operation to obtain the final output. The k-fold ensemble approach is shown in Figure 2.

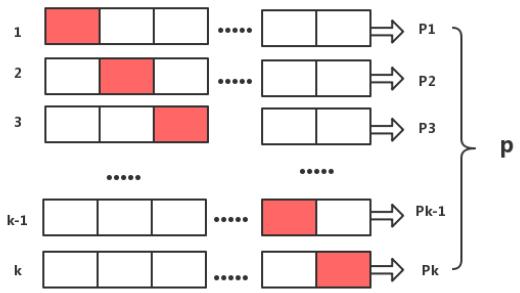

Fig. 2. The k-fold ensemble approach

The purpose of this operation is that different data sets will be trained during each training process, and different features will be extracted during the process of extracting features from the model, which can further improve the generalization ability of the model.

4 Experiment and results

4.1 Data preprocessing

Whether in official dataset or the OLID dataset, from Twitter or Facebook, the data is very noisy because it is not processed. Tweets are first processed using the Tweetokenize tool [1]. In order to extract the model into better features, we further process the data, the specific steps are as follows:

- The word is retained for hashtags. Because it is very unique in itself, and it may help with text categorization.
- Username mentions, e.g.: words starting with "@", they are replaced with 'username'. Because we think that usernames don't contain emotional expressions, and various usernames can bring a lot of unknown words, which can lead to reduced model performance.
- All contractions are split into two tokens(e.g.: "you're" is changed to "you" and "are").
- The emojis are replaced with the corresponding words by emotion lexicons[2] to express the corresponding emotion.
- Lemmatization is restored to general form by WordNetLemmatizer.
- Tokens are converted to lower case.

[1] https://www.github.com/jaredks/tweetokenize
[2] https://emojipedia.org/

Since the official verification set is not provided, we randomly extracted 500 tweets from the OLID data set and the officially released training set as the verification set, and the ratio of NOT and HOF in the verification set is 1:1.

4.2 Experiment setting

In our model, for this encoding layer, we set the hidden units to 128 and num levels to 16; We added a layer of dropout between the encoding layer and the attention layer. The purpose of this layer is to improve the generalization of the model and prevent the model from overfitting. Connected behind the Attention layer is a layer of Dense with Relu activation function, and the number of hidden units is 128. Finally, we added the Dropout layer and the BatchNormalization layer, and the rate of all Dropout layers is 0.25. The activation function of the final output layer is sigmoid for binary classification. The loss function of this model is binary crossentropy, and the optimizer is adam.

By using 5-fold crossvalidation on the training data, we set the batch size to 512 and the epoch to 20 for training. And the pre-training word vector we used is fastText, which is provided by Mikolov et al. [7]. It is a 2 million word vector trained using subword information on Common Crawl with 600B tokens, and its dimension is 300.

4.3 Result

This Subtask A is to evaluate the classification system by calculating the Marco F1 score and Weighted F1 score. According to the official results in Subtask A for English language, the best Marco F1 score and Weighted F1 score of our model are 0.7882 and 0.8395 respectively, ranked 1st place, and our result is 0.0188 higher than the second place for Marco F1 score. The three results we submitted are shown in Table 1, and the top 5 teams from the official leaderboard is shown in Table 2.

Run	**Macro F1**	**Weighted F1**
1	0.7682	0.8175
2	**0.7882**	**0.8395**
3	0.772	0.8237

Table 1. The three results we submitted in Subtask A for English language

Run 1 is the result without using the k-fold ensemble method, while run 2 used the k-fold ensemble method. It can be seen that the model's performance is increased by 0.02 for run 2, which is very effective and provides more powerful performance for the model. The parameters of run 3 are only slightly different from those of run 2.

From the detailed results provided by the official, our model's F1-score for HOF is only 0.69, which is much lower than the 0.89 of NOT, indicating that

Team	Macro F1	Weighted F1
YNU_wb	**0.7882**	**0.8395**
BRUMs	0.7694	0.838
vito	0.7568	0.8182
3Idiots	0.7465	0.8012
IIITG-ADBU	0.7462	0.8064

Table 2. Top-5 for the official leaderboard in Subtask A for English language

our model excels at distinguishing between NOT, which may be due to training data. The number of HOFs is lower than NOT, and we have not dealt with the imbalance of data. The detailed results of the run_2 are shown in Table 3.

Label	precision	recall	f1-score
HOF	0.66	0.71	0.69
NOT	0.90	0.88	0.89

Table 3. The detailed results of the run_2

5 Conclusion

In this paper, we presented a model to identify hate speech and offensive language for English language, and also used the k-fold ensemble method to improve the generalization ability of the model, and achieved the best results in Subtask A for English language. In future research, we will consider handling data imbalances and introducing location features to try to further enhance the performance of the model.

Acknowledgments

This work was supported by the Natural Science Foundations of China under Grants 61463050, the NSF of Yunnan Province under Grant 2015FB113.

References

1. Elizabeth Whittaker and Robin M Kowalski. 2015. Cyberbullying via social media. Journal of School Violence, 14(1):11C29.
2. Sandip Modha, Thomas Mandl, Prasenjit Majumder and Daksh Patel. 2019. Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation.

3. Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, 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, pages 54-63.
4. Paula Fortuna and Srgio Nunes. 2018. A Survey on Automatic Detection of Hate Speech in Text. ACM Computing Surveys (CSUR), 51(4):85.
5. Vijayasaradhi Indurthi, Bakhtiyar Syed, Manish Shrivastava, Nikhil Chakravartula, Manish Gupta, and Vasudeva Varma. 2019. Fermi at semeval-2019 task 5: Using sentence embeddings to identify hate speech against immigrants and women in Twitter. In Proceedings of the 13th International Workshop on Semantic Evaluation, pages 70C74.
6. Ping Liu, Wen Li, and Liang Zou. 2019. Nuli at semeval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In Proceedings of the 13th International Workshop on Semantic Evaluation, pages 87C91.
7. Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)
8. Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2017. Hierarchical attention networks for document classification. In Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1480C1489.
9. Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pages 1C11.
10. Montani, Joaquın Padilla.2018. Tuwienkbs at germeval 2018: German abusive tweet detection. In14thConference on Natural Language Processing KONVENS 2018, page 45.
11. Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2018. Ordered neurons: Integrating tree structures into recurrent neural networks. arXiv preprint arXiv:1810.09536.
12. Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In Proceedings of GermEval.
13. 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.
14. 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).