QMUL-NLP at HASOC 2019: Offensive Content Detection and Classification in Social Media

Aiqi Jiang

Queen Mary University of London, London E1 4NS, UK
aiqi.jiang@yahoo.com

Abstract. With the development of the Internet, the Web has become an information dissemination platform, an information amplifier, and a new social media. The information load and participation of the Internet far exceeds the existing traditional media, and various problems have emerged. There has been significant work in several languages in particular for English. However, there is a lack of research in this recent and relevant topic for most other languages. This track intends to develop data and evaluation resources for several languages. The objectives are to stimulate research for these languages and to find out the quality of hate speech detection technology in other languages. The paper mainly describes the organization of the HASOC 2019 Task, a Shared Task on Hate Speech and Offensive Content Identification in Indo-European Languages. The task is organized in three related classification subtasks: subtask A is a coarse-grained binary classification to identify hate speech and offensive language, a fine-grained classification subtask B is to further classify the data from the subtask A into three categories, and subtask C will check the type of offense. This paper mainly focuses on English offensive language detection and shows the experimental result in subtask A and subtask B.

Keywords: Hate speech detection · Offensive language · Word embedding · Text classification · LSTM · HASOC.

1 Introduction

With the popularity of Internet applications and the convenience of free speech, a lot of hate speech and other offensive content on the Internet pose a huge threat to the stability of society. The online communication platform has no strict scrutiny of speech and post, making a variety of offensive language, such as insulting, harmful, derogatory or obscene, freely and quickly transmitted from person to person, and can have an influence on people's views and social trends

[17]. A large number of such languages contain many of these critical and misleading statements and lack factual research, which may lead to some excessive behavior in society and pose a threat to democracy. Therefore, in order to prevent the abuse and transmission of hate speech on social media, the accurate detection of hate speech is urgent. At present, many online communities, social media companies and technology companies pay great attention to this related research, investing a lot of money and technical support [16].

The main structure of the rest of this paper has been organized as follows. Section 2 will mainly describe recent researches about hate speech detection. Then the datasets released by HASOC to the participants for training and testing the systems will be introduced in section 3. Section 4 presents the two subtasks and the measures we exploited in the evaluation. Section 5 reports on approaches in the experiment and results of the system. Finally a conclusion will be given in section 6.

2 Related Work

Existing works on hate speech has been very limited, largely due to a lack of a general definition of hate speech, a lack of analysis of its demographic impact, and a lack of surveys of the most effective characteristics [11]. Generally speaking, hate speech is based on attacks on individuals or groups in certain ways, such as gender, race, religion, ethnicity, disability or sexual orientation. It means deliberately suppressing, intimidating, or inciting some statements about violence and prejudice against individual groups [15].

Related researches on hate speech detection has been developing only in recent a few years. The existing technology used in hate speech detection in social media is mainly about Dictionaries and lexicons [4], Bag-of-words(BOW) [3], TF-IDF [1], Part-of-speech(POS) [3], and Word embedding [2]. Many recent studies have shown that deep learning techniques with word embedding show higher accuracy in text categorization [2]. Among them, Word2Vec has obtained many applications [7], a method based on unsupervised word embedding to find the semantic and syntactic relationships of words to then capture the more attributes and contextual hints in human language..

The most common method found in the work of [6] is to establish a machine learning model for hate speech classification. Considering the discovery frequency, the most commonly used algorithms are SVM, Random Forests, Decision Trees, Logistic Regression and Naive Bayes, where Random Forests and Logistic Regression show a good performance. As for deep learning methods, existing ones are largely based on Convolutional Neural Networks (CNN) or Long Short Term Memory (LSTM), a type of Recurrent Neural Networks (RNN) [14]. Intuitively, traditional machine learning methods learn features similar to n-gram sequences, while deep learning ones learn sequence order, which seems more useful for classification tasks [9].

In this paper, we choose to use English dataset to mainly address Subtask A and Subtask B, where different feature extraction methods (N-gram and word embedding) and classification algorithms (Logistic Regression and LSTM) will be implemented in this experiment.

3 Data

3.1 Datasets

The training dataset provided by HASOC is created mainly from the Twitter and Facebook in English. It is raw data with text ID number, post content and different class labels for three subtasks. The external dataset used the public Twitter search API to collect the entire corpus, filtering for tweets not written in English [13]

The size of English training data corpus is 5852 posts and the external dataset has 39292 texts. The size of test dataset has 1153 posts. The following Table 1 shows the details of three datasets we used.

Table 1. Size of data sets for English

Datasets	Number of the text
HASOC training data	5852
Annotated data	39292
HASOC test data	1153

3.2 Training and Test Data

- **Training data**
  Training data set has randomly combined the training dataset HASOC released and the external dataset. Then the combined dataset is divided into a training set and a test set according to the ratio of 4:1.

- **Test data**
  The test data set is what HASOC released with approximately 1100 posts.

4 Task description

The format of an annotated text in the training and development set shows the pattern as follows:

ID, text, task_1, task_2, task_3

where ID is a progressive number denoting the text within the dataset, text is the given post, while the other three parts of the pattern are the labels of the classes for the texts. And the test set only includes ID and text.

An example of one post is as follows:

hasoc_en_2, @politico No. We should remember very clearly that Individual1 just admitted to treason. TrumpIsATraitor McCainsAHero JohnMcCainDay, HOF, HATE, TIN

where the text has been classified by the annotators as hateful-offensive, hateful, and insulting to an individual, group, or others.

4.1 Subtask A

Subtask A is a coarse-grained binary classification task to make Hate speech and Offensive language identification [5]. The system has to predict whether a text in English contains hate speech and offensive information or not.

For the class of this subtask, there is two labels: HOF and NOT. The label HOF means it contains any form of non-acceptable language such as hate speech, aggression, profanity otherwise NOT.

4.2 Subtask B

Subtask B is a fine-grained multi-level classification task to further identify three classes: HATE, OFFN and PRFN. There are four annotations, where most of posts is classified to OTHER, some to be HATE and the other two categories to be relatively less. Dubious cases, which are difficult to decide even for humans, will be left out [5].

4.3 Evaluation Measures and Baseline

In the result of binary classification, there are four different situations, namely true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Based on the results of manual annotations, there are four commonly used indicators to measure the performance of the classifier, namely accuracy, recall and F1 scores [12].

- **Precision**
  positive predictive value: it is a consistent result between manual and automatic classification.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

- **Recall**
  sensitivity: it shows the proportion of all positive cases, which is a measure of the ability of the classifier to identify positive samples.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

In this report, the evaluation measures are the same for both subtask A and subtask B. To provide a metric that is independent of class size, the classification result will be mainly computed by macro-averaged F1 score and weighted F1 score [10], which are based on metrics mentiond above.

- **Macro-averaged F1 score**
  It is firstly calculated for each category of indicator values, and then for the arithmetic mean of all categories.

$$Macro_P = \frac{\sum Precision}{n} \qquad Macro_R = \frac{\sum Recall}{n} \tag{3}$$

$$Macro_F = \frac{2 \times Macro_P \times Macro_R}{Macro_P + Macro_R} \tag{4}$$

- **Weighted F1 score**
  It is firstly calculated for each label and then averaged by support weighting - the actual number of instances per label.

$$Weighted_P = \frac{\sum TP}{\sum TP + \sum FP} \qquad Weighted_R = \frac{\sum TP}{\sum TP + \sum FN} \tag{5}$$

$$Weighted_F = \frac{2 \times Weighted_P \times Weighted_R}{Weighted_P + Weighted_R} \tag{6}$$

5 Participant Systems and Result

The hate speech detection system is implemented in the process of four parts, namely text preprocessing, fearture extraction, classifier building, and classification. Then the classification results from different models will be analyzed.

5.1 Experiments

Text preprocessing The text usually contains a lot of meaningless or unaffected information that may affect research results at different stages, such as punctuation, common words, links, and numbers. In this step, regular expressions have been used to eliminate noise, including non-alphanumeric characters and numbers. And we remove text information noise like stopwords as well, which are probably of little value in hate speech detection later. The stop words list in NLTK corpus has been chosen to delete meaningless words in texts. Besides, the post in social media can commonly include many non-point content, such as the mention to user, specific topic and URL links. So these contents is replaced by the corresponding words, namely USER, TOPIC and URL.

Feature extraction Before training the model, it is necessary to convert the text to various feature vectors because the preprocessed text cannot be directly recognized by the model. In this step, I mainly consider trying to use two common features: n-gram and word embedding features. They can be compared according to the final results generated by the classifier.

- **N-gram feature**
  I mainly focus on unigram features, and then select bag-of-words (BOW) model for n-gram feature notation. It is fairly straighforward and each element demonstrates how often the term appears in a text sentence. Since the information of the low frequency words is more abundant, I use the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm to convert the frequency into the weight of the word, which is a quite robust and accurate weight calculation method. Feature processing can be done through CountVectorizer() and TfidfTransformer() in the NLTK package.
- **Word embeddings feature**
  The vector dimension is low and dense, and the information density is high in word embedding. We will use Word2Vec model, where the similarity between words can be directly reflected by the calculation of word vectors. In order to create the Word2Vec feature, it is decided to use the pre-trained Google Word2Vec model, which has more convincing information about word similarity. Then the similarity value of each word will be the average of its word embedded value in the Word2Vec feature vector.

Classifier building The whole experimental process is able to be achieved by NLTK, scikit-learn and the Keras system. The same two feature will be the input feature data for different classifiers for subtask A and subtask B.

- **Subtask A**
  For both n-gram feature and word embedding feature, traditional machine learning algorithm Logistic Regression (LR) and deep learning sequential algorithm Long Short-Term Memory(LSTM) are respectively implemented as a binary classifier.
- **Subtask B**
  The One-vs-all classifier will be built by using LR and LSTM for both two extracted features.

Classifcation There will be four different classification experiments. Three of them will implement experiments on HASOC test data initially given by HASOC organizers, where Logistic Regression classifier will input TF-IDF and Word2Vec features respectively, and LSTM model will utilize Word2Vec feature. The other experiment is implemented by providing HASOC organizers with our LSTM model with Word2Vec feature, which the final result is based on a new test dataset used privately by HASOC organizers. HASOC test will show the final F1 score result from HASOC organizers.

5.2 Result Analysis

The feature dimensionality of one-hot representation is rather high, which is easy to lead to a poor training model. So the LR classifier with TF-IDF feature is considered as the baseline for comparison.

- **Subtask A**
  Our result is ranked in 9th position, seemingly a good score. There is not a big difference between two F1 scores. The LSTM classifier with word embedding features has the best performance.

Table 2. The result of Subtask A

	TF-IDF + LR	Word2Vec + LR	Word2Vec + LSTM	HASOC test
macro F1	0.7991	0.7793	**0.8104**	0.7431
weighted F1	**0.8738**	0.8435	0.8661	0.8163

- **Subtask B**
  The HASOC result is ranked in 32th, a not very good result. The weighted F1 values show a good performance, but it has a big difference from the macro-averaged F1 values.

Table 3. The result of Subtask B

	TF-IDF + LR	Word2Vec + LR	Word2Vec + LSTM	HASOC Test
macro F1	0.3029	0.2598	**0.3083**	0.2740
weighted F1	0.6738	0.6032	**0.6955**	0.6807

It can be seen that the LR classifier using pre-trained word embedding model do not work much better than the classifier using TF-IDF feature, which may be because the Google pre-trained word embedding model is based on the field of news instead of Twitter. And because of this, the performance of the LSTM model is not particularly good in the final result.

6 Conclusion

The spread of hate speech on social media has increased significantly in recent years, which could have a serious effect on the society. Therefore, our work makes several contributions according to this problem. First, we try several methods classifying hate speech using both traditional machine learning model like LR and deep learning model like LSTM, to empirically improve classification accuracy. Second, we create a new hate speech dataset by combining an external dataset together with the original released one from HASOC organizers. Third, the pre-trained model for word embedding feature extraction is used to improve the accuracy of hate speech classification. Our results show a good performance in both two F1 scores in Subtask A and weighted F1 score in Subtask B, while subtask B needs a further fine-grained experiment based on specific classes.

References

1. Agarwal, S., Sureka, A. (2017). Characterizing linguistic attributes for automatic classification of intent based racist/radicalized posts on tumblr micro-blogging website. *arXiv preprint arXiv:1701.04931.*
2. Badjatiya, P., Gupta, S., Gupta, M., Varma, V. (2017, April). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion* (pp. 759-760). International World Wide Web Conferences Steering Committee.
3. Davidson, T., Warmsley, D., Macy, M., Weber, I. (2017, May). Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media.*
4. Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., Bhamidipati, N. (2015, May). Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web* (pp. 29-30). ACM.
5. Modha, S., Mandl, T., Majumder, P., Patel, D. (2019, December). 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.*
6. Mehdad, Y., Tetreault, J. (2016, September). Do characters abuse more than words?. In Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue (pp. 299-303).
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).
8. Mossie, Z., Wang, J. H. (2018). Social Network Hate Speech Detection for Amharic Language. *Computer Science Information Technology*, 41
9. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y. (2016, April). Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web* (pp. 145-153). In International World Wide Web Conferences Steering Committee.
10. Ozgur, A., Ozgur, L., Gungor, T. (2005, October). Text categorization with class-based and corpus-based keyword selection. In *International Symposium on Computer and Information Sciences*(pp. 606-615). Springer, Berlin, Heidelberg.
11. Schmidt, A., Wiegand, M. (2017, April). A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media* (pp. 1-10).
12. Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1), pp. 1-47.
13. Waseem, Z., Hovy, D. (2016, June). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*(pp. 88-93).
14. Wei, X., Lin, H., Yang, L., Yu, Y. (2017). A convolution-LSTM-based deep neural network for cross-domain MOOC forum post classification. *Information*, 8(3), 92.
15. Wikipedia page about hate speech, https://en.wikipedia.org/wiki/Hate_speech. Last accessed 12 Oct 2019
16. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R. (2019). Predicting the Type and Target of Offensive Posts in Social Media. arXiv preprint arXiv:1902.09666.

17. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R. (2019). Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.