USF at SemEval-2019 Task 6: Offensive Language Detection Using LSTM With Word Embeddings

Bharti Goel and Ravi Sharma
Department of Computer Science and Engineering, University of South Florida, USA
Emails: bharti@mail.usf.edu, ravis@mail.usf.edu

Abstract

In this paper, we present a system description for the SemEval-2019 Task 6 submitted by our team. For the task, our system takes tweet as an input and determine if the tweet is offensive or non-offensive (Sub-task A). In case a tweet is offensive, our system identifies if a tweet is targeted (insult or threat) or non-targeted like swearing (Sub-task B). In targeted tweets, our system identifies the target as an individual or group (Sub-task C). We used data pre-processing techniques like splitting hashtags into words, removing special characters, stop-word removal, stemming, lemmatization, capitalization, and offensive word dictionary. Later, we used keras tokenizer and word embeddings for feature extraction. For classification, we used the LSTM (Long short-term memory) model of keras framework. Our accuracy scores for Sub-task A, B and C are *0.8128*, *0.8167* and *0.3662* respectively. Our results indicate that fine-grained classification to identify offense target was difficult for the system. Lastly, in the future scope section, we will discuss the ways to improve system performance.

1 Introduction

In recent years, there has been a rapid rise in social media platforms and surge in the number of users registering in order to communicate, publish content, showcase their skills and express their views. Social media platforms like Facebook and Twitter have millions of registered users influenced by the countless user-generated posts on daily basis (Zeitel-Bank and Tat, 2014). While on one hand social media platforms facilitate the exchange of views, effective communication and can be seen as a helping mode in crisis. On the other hand, they open up the window for anti-social behavior such as bullying, stalking, harassing, trolling and hate speech (Malmasi and Zampieri, 2018; Wiegand et al., 2018; ElSherief et al., 2018; Zhang et al., 2018). These platforms provide the anonymity and hence aid users to indulge in aggressive behavior which propagates due to the increased willingness of people sharing their opinions (Fortuna and Nunes, 2018).

This aggression can lead to foul language which is seen as "offensive", "abusive", or "hate speech", terms, which are used interchangeably (Waseem et al., 2017). In general, offensive language is defined as derogatory, hurtful/ obscene remarks or comments made by an individual (or group) to an individual (or group) (Wiegand et al., 2018; Baziotis et al., 2018). The offensive language can be targeted towards a race, religion, color, gender, sexual orientation, nationality, or any characteristics of a person or a group. Hate Speech is slowly plaguing the social media users with depression and anxiety (Davidson et al., 2017; Zhang et al., 2018), which can be presented in the form of images, text or media such as audio, video, etc. (Schmidt and Wiegand, 2017).

Our paper presents the data and task description followed by results, conclusion and future work. The purpose of Task 6 is to address and provide an effective procedure for detecting offensive tweets from the data set provided by shared task report paper (Zampieri et al., 2019b). The shared task is threefold. The Sub-task A ask us to identify whether the given tweet is offensive or non-offensive. In Sub-task B offensive tweets are to be classified as targeted (person/group) or non-targeted (general). Sub-task C ask us to do classification of the offensive tweets into individual, group or others. We apply the LSTM with word embeddings in order to perform the multi-level classification.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 796–800
Minneapolis, Minnesota, USA, June 6–7, 2019.

2 Related Work

Technological giants like Facebook, Google, YouTube and Twitter have been investing a significant amount of time and money towards the detection and removal of offensive and hate speech posts that give users a direct or indirect negative influence (Fortuna and Nunes, 2018). However, lack of automation techniques and ineffectiveness of manual flagging has lead to a lot of criticism for not having potent control for the problem (Zhang et al., 2018). The process of manual tagging is not sustainable or scalable with the large volumes of data exchanged in Social media. Hence, the need of the hour is to do automatic detection and filtering of offensive posts to give the user quality of service (Fortuna and Nunes, 2018).

The problem of automatic Hate Speech Detection is not trivial as offensive language may or may not be meant to insult or hurt someone and can be used in common conversations. Different language contexts are rampant in social media (Davidson et al., 2017). In recent years, linguistics, researchers, computer scientists, and related professionals have conducted research towards finding an effective yet simple solution for the problem. In papers, (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018), authors survey the state of the art methods along with the description of the nature of hate speech, limitations of the methods proposed in literature and categorization of the hate speech. Further, authors mainly classified the features as general features like N-grams, Part-Of-Speech (POS) and sentiment scores and, specific features such as othering language, the superiority of in-group and stereotype. In (Silva et al., 2016), authors list categories of hate speech and possible targets examples.

The research carried over the years has employed various features and classification techniques. The features include the bag of words (Greevy and Smeaton, 2004; Kwok and Wang, 2013), dictionaries, distance metrics, N-grams, IF-IDF scores, and profanity windows (Fortuna and Nunes, 2018). Authors in (Davidson et al., 2017) used a crowdsourced hate speech lexicon to collect tweets and train a multi-class classifier to distinguish between hate speech, offensive language, and non-offensive language. The authors in paper (Waseem et al., 2017) presents a typology that gives the relationship between various sub-tasks such as cyber-bullying, hate speech, offensive, and online abuse. They synthesize the various literature definitions and contradiction together to emphasize the central affinity among these sub-tasks. In (Gambäck and Sikdar, 2017), authors used a Convolutional Neural Network model for Twitter hate speech text classification into 4 classes: sexism, racism, both sexism-racism and neither. Similar approaches using deep learning have been employed in (Agrawal and Awekar, 2018; Badjatiya et al., 2017) for detecting hate speech and cyberbullying respectively.

Authors in (Zhang et al., 2018) have proposed Deep Neural Network (DNN) structures which serve as a feature extractor for finding key semantics from hate speech. Prior to that, they emphasize the linguistics of hate speech, that it lacks discriminative features making hate speech difficult to detect. Authors in (ElSherief et al., 2018) have carried out the analysis and detection of the hate speech by classifying the target as directed towards a specific person/identity or generalized towards a group of people sharing common characteristics. Their assessment states that directed hate consists of informal, angrier and name calling while generalized hate consists of more religious hate and use of lethal words like murder, kills and exterminate.

3 Problem Statement

In Task 6, three level offensive language identification is described as three Sub-tasks A, B and C. For Sub-task A, tweets were identified as offensive (tweet with any form of unacceptable language, targeted/non-targeted offense) or not offensive. For Sub-task B, the offensive tweets are further categorized into targeted or non-targeted tweets. The targeted offense is made for an individual or group while the untargeted offense is a general use of offensive language. Later for Subtask C, targeted tweets are further categorized according to the target, individual, group or others. This step by step tweet classification will lead to the detailed categorization of offensive tweets.

4 Data

The data collection methods used to compile the dataset used in OffensEval are described in Zampieri et al. (2019a). The OLID dataset collected from Twitter has tweet id, tweet text, and

labels for Sub-task A, B, and C. We have also used Offensive/Profane Word List (Ahn, 2019) with *1,300+* English terms that could be found offensive for lexical analysis of tweets to see check probability of tweet being offensive if a tweet has an offensive word.

4.1 Data Pre-processing

The data is raw tweet data from Twitter and hence data cleaning and pre-processing is required. We used the following steps for data pre-processing:

1. Split hashtags over Capital letters: In this step hashtags are divided into separate words (for example, "#TweetFromTheSeat" will be converted to "Tweet from the seat"). Generally, while writing hashtags multiple words are combined to form single hashtag, where each word is started with a capital letter. Here, we take advantage of this property of hashtag and generate a string of words from it.

2. Remove special characters: In this step we removed all special characters from the tweet and resultant tweet will contain only alphabets and number. In Twitter domain "#" is important special character. Splitting of hashtags,"#Text" into "#" and "Text", retains the purpose of the hashtags after removal of "#" . Other special characters (for e.g. ",", ".","!", etc) are not much informative for given context.

3. Removal of stop-words, Stemming and Lemmatization: In this step we used NLTK (Loper and Bird, 2002) list of stop-words to remove stopwords, classic Porter Stemmer (Porter, 1980) for stemming and NLTK (Loper and Bird, 2002) Word Net Lemmatizer for lemmatization.

4. Capitalization: This is the last step for data pre-processing and all characters are converted to capital letters. In Twitter domain uppercase characters are said to portray expression, but this is not true for all cases. Also, keeping cases intact may lead to over-fitting during training.

5. Embedding "Offensive": This is an optional step. We used offensive word list (Ahn, 2019) to find offensive words in the tweet. Later for tweets with matched offensive words were embedded with "Offensive" as a word.

5 System Description

We used a four-layer neural network with each layer detailed below:

System	F1 (macro)	Accuracy
All NOT baseline	0.4189	0.7209
All OFF baseline	0.2182	0.2790
LSTM (5,0.2)	**0.7382**	**0.8128**

Table 1: Results for Sub-task A.

System	F1 (macro)	Accuracy
All TIN baseline	0.4702	**0.8875**
All UNT baseline	0.1011	0.1125
LSTM (5,0.2)	0.5925	0.8167
LSTM (20,0.2)	0.5291	0.6125
LSTM (20,0.2) and word list	**0.6171**	0.7667

Table 2: Results for Sub-task B.

The first layer of our network is an embedding layer, which takes tokens as inputs (each sentence is converted into index sequences using tokenizer). This layer convert sequences to dense vector sequences generating embedding table used by the next layer. We used tokenizer for top *1000* words and embedding dimension of *128* for our system. The second layer is SpatialDropout1D layer which helps promote independence between feature maps. We used *0.1* as rate or Fraction of the input units to drop. This layer is mainly used to covert multi-dimensional input to one-dimensional input using dropout method.

Third layer is LSTM (Hochreiter and Schmidhuber, 1997) layer with dropout and recurrent dropout as *0.2*. This layer serves as a recurrent neural network layer which was only for short term memory. LSTM (long short-term memory) takes care of longer dependencies. The dimension of LSTM hidden states is *200* for our system. Finally, we used a dense layer with Softmax function for binary classification in-case of Sub-task A and B, and three class classification for Sub-task C. The dimension of the dense layer is *200*.

For hyperparameter selection, we used different train and validation splits. The batch size is *64*, and the maximal training epoch is varied with different system ranging from *5* to *50* (Performance was decreasing for higher epochs). We used RMSProp as the optimizer for network training. The performance is evaluated by macro-averaged F1-score and Accuracy by task organizers.

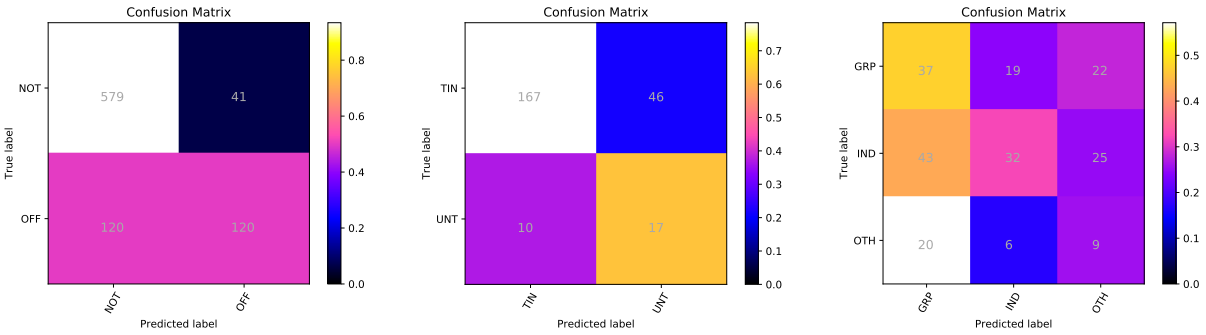

Figure 1: a) Sub-task A, LSTM (epoch=5, dropout= 0.2), b) Sub-task B, LSTM (dropout=0.2, epochs=20), c) Sub-task C, LSTM(dropout=0.2, epochs=50)

System	F1 (macro)	Accuracy
All GRP baseline	0.1787	0.3662
All IND baseline	0.2130	**0.4695**
All OTH baseline	0.0941	0.1643
LSTM (20,0.2) and word list	0.1849	0.1878
LSTM (50,0.2)	**0.3404**	0.3662
LSTM (50,0.2) and word list	0.2832	0.3521

Table 3: Results for Sub-task C.

6 Results

Tables 1, 2 and 3 shows F1 (macro) and Accuracy scores for the submitted systems. We can see that for all the sub-tasks our system has F1 (macro) and as described in (Zampieri et al., 2019a) F1 (macro) is used for performance analysis by task coordinators. Best results are highlighted in bold ink in tables and confusion matrix for them is also shown in Figure 1 for Sub-tasks A, B and C. In **Sub-task A** we achieved *0.8128* and *0.7382* as accuracy and F1 respectively. For this task we submitted only one system with LSTM network dropout *0.2* and *5* epochs. For **Sub-task B** we submitted three runs but, the best performance is achieved by system with LSTM dropout of *0.2*, *20* epochs and offensive word list described in Section 4. Later in **Sub-task C** we submitted three runs and best performance was LSTM with *50* epochs and *0.2* dropout.

7 Conclusion

In this paper we used LSTM network to identify offensive tweets and categorize offense in subcategories as described in Section 3 for Task 6: Identifying and Categorizing Offensive Language in Social Media. We used an embedding layer followed by LSTM layer for tweet classification. Three tasks of OffensEval Sub-task A, B, and C were of varied difficulty level. The main reason can be decreasing amount of data for each of them, where Sub-task A has more data followed by Sub-task B categorizing offensive tweets identified by Sub-task A and, Sub-task C categorizing targeted offense identified by Sub-task B. Data was also unbalanced leading to more importance for majority class but after applying cost function we found that accuracy was decreased with increased errors in identification of majority class.

8 Future Work

For future work, we would like to use additional datasets like TRAC-1 data (Kumar et al., 2018), (Davidson et al., 2017), and would collect data from Twitter to get diverse data. To be consistent with substantial research done in recent years we want to employ a combination of textual features like the bag of words n-grams, capitalized characters, sentiment scores, e.t.c. Also, we want to focus more on specific features like semantics and linguistic features intrinsic to hate/offensive rather than just generic text-based features. For that, we want to use character level deep LSTM which can be used to extract the semantic and syntactic information. Finally, we want to explore more about the similarities and dissimilarities between the profanity and hate speech, establishing more profound way of extracting features in order to make the detection system more responsive.

References

Sweta Agrawal and Amit Awekar. 2018. Deep learning for detecting cyberbullying across multiple social media platforms. *CoRR*, abs/1801.06482.

Luis von Ahn. 2019. Offensive/profane word list. *Carnegie Mellon School of Computer Science*.

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 Companion*, WWW '17 Companion, pages 759–760, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Baziotis, Christos, Athanasiou, Nikos, Athanasia, Paraskevopoulos, Georgios, Ellinas, Nikolaos, Alexandros, and et al. 2018. Ntua-slp at semeval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive rnns.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of ICWSM*.

Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding. 2018. Hate Lingo: A Target-based Linguistic Analysis of Hate Speech in Social Media. *arXiv preprint arXiv:1804.04257*.

Paula Fortuna and Sérgio Nunes. 2018. A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys (CSUR)*, 51(4):85.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using Convolutional Neural Networks to Classify Hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90.

Edel Greevy and Alan F. Smeaton. 2004. Classifying racist texts using a support vector machine. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 468–469, New York, NY, USA. ACM.

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

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 Cyberbulling (TRAC)*, Santa Fe, USA.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting Tweets Against Blacks. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:1–16.

M.F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection Using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics*, pages 1–10, Valencia, Spain.

Silva, Leandro, Mondal, Correa, Denzil, Benevenuto, Fabricio, Weber, and Ingmar. 2016. Analyzing the targets of hate in online social media.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In *Proceedings of the First Workshop on Abusive Langauge Online*.

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*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

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)*.

Natascha Zeitel-Bank and Ute Tat. 2014. *Social Media and Its Effects on Individuals and Social Systems*, Human Capital without Borders: Knowledge and Learning for Quality of Life; Proceedings of the Management, Knowledge and Learning International Conference 2014, pages 1183–1190. ToKnowPress.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *Lecture Notes in Computer Science*. Springer Verlag.