The Titans at SemEval-2019 Task 6: Offensive Language Identification, Categorization and Target Identification

Avishek Garain
Computer Science and Engineering
Jadavpur University, Kolkata
avishekgarain@gmail.com

Arpan Basu
Computer Science and Engineering
Jadavpur University, Kolkata
arpan0123@gmail.com

Abstract

This system paper is a description of the system submitted to "SemEval-2019 Task 6", where we had to detect offensive language in Twitter. There were two specific target audiences, immigrants and women. The language of the tweets was English. We were required to first detect whether a tweet contains offensive content, and then we had to find out whether the tweet was targeted against some individual, group or other entity. Finally we were required to classify the targeted audience.

1 Introduction

Offensive language is pervasive in social media. Individuals frequently take advantage of the perceived anonymity of computer-mediated communication, using this to engage in behavior that many of them would not consider in real life. Online communities, social media platforms, and technology companies have been investing heavily in ways to cope with offensive language to prevent abusive behavior in social media.

One of the most effective strategies for tackling this problem is to use computational methods to identify offense in user-generated content (e.g. posts, comments, microblogs, etc.). This topic has attracted significant attention in recent years of various Natural Language analysts.

The SemEval 2019 task 6 (Zampieri et al., 2019b) was a classification task where we were required to classify a tweet, as hate speech or otherwise. However, there were some additional challenges presented, which involved automatic categorization of offense target types and the specific detection of the target audience, namely, women or immigrants.

The task was divided into three parts. In the first subtask our system categorized the instances into `OFF` and `NOT`. In the second subtask our system categorized instances into `TIN` and `UNT` while in the third subtask systems should categorize instances into `IND`, `GRP`, and `OTH`.

To solve the task in hand we built a bidirectional LSTM based neural network for prediction of the classes present in the provided dataset.

The paper has been organized as follows. Section 2 describes a brief survey on the relevant work done in this field. Section 3 describes the data, on which, the task was performed. The methodology followed is described in Section 4. This is followed by the results and concluding remarks in Section 5 and 6 respectively.

2 Related Work

Papers published in the last two years include the surveys by Schmidt and Wiegand (2017) and Fortuna and Nunes (2018). The paper by Davidson et al. (2017) presenting the Hate Speech Detection dataset were used in (Malmasi and Zampieri, 2017) and a few other recent papers such as (ElSherief et al., 2018; Gambäck and Sikdar, 2017; Zhang et al., 2018).

A proposal of typology of abusive language sub-tasks is presented in (Waseem et al., 2017). For studies on languages other than English see work by Su et al. (2017) on Chinese and Fišer et al. (2017) on Slovene. Finally, for recent discussion on identifying profanity vs. hate speech see the work by Malmasi and Zampieri (2018). This work highlighted the challenges of distinguishing between profanity, and threatening language which may not actually contain profane language.

Previous editions of related workshops are TA-COS[1], Abusive Language Online[2], and TRAC[3] and related shared tasks such as GermEval (Wiegand et al., 2018) and TRAC (Kumar et al., 2018).

[1] http://ta-cos.org/
[2] https://sites.google.com/site/abusivelanguageworkshop2017/
[3] https://sites.google.com/view/trac1/home

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 759–762
Minneapolis, Minnesota, USA, June 6–7, 2019.

3 Data

The dataset that was used to train the model is the OLID dataset (Zampieri et al., 2019a). It was collected from Twitter; the data being retrieved the data using the Twitter API by searching for keywords and constructions that are often included in offensive messages. The vast majority of content on Twitter is not offensive so different strategies were tried to keep a reasonable number of tweets in the offensive class amounting to around 30% of the dataset.

The dataset provided consisted of tweets in their original form along with the corresponding labels. Subtask A consisted of the labels `OFF` and `NOT`; subtask B consisted of the labels `TIN` and `UNT`; and finally subtask C consisted of the labels `IND`, `GRP` and `OTH`.

Label	Meaning
OFF	Tweet containing offensive language
NOT	Tweet not containing offensive language
TIN	Tweet containing profanity and targeted against individual/group/others
UNT	Tweet with profanity, but non-targeted
IND	Offensive tweet targeting an individual
GRP	Offensive tweet targeting a group
OTH	Offensive tweet targeting neither group or individual

Table 1: Meaning of the labels used in the dataset

The dataset had 14100 instances which were divided into 13240 training data instances and 860 test data instances.

A	B	C	Train	Test	Total
OFF	TIN	IND	2407	100	2507
OFF	TIN	OTH	395	35	430
OFF	TIN	GRP	1074	78	1152
OFF	UNT	-	524	27	551
NOT	-	-	8840	620	9460
All			13240	860	14100

Table 2: Distribution of the labels in the dataset

4 Methodology

Our approach was to convert the tweet into a sequence of words and then run a neural-network based algorithm on the processed tweet.

The first stage in our pipeline was to preprocess the tweet. This consisted of the following steps:

1. Removing mentions
2. Removing punctuation
3. Removing URLs
4. Contracting white space
5. Extracting words from hash tags

The last step consists of taking advantage of the Pascal Casing of hash tags (e.g. `#PascalCasing`). A simple regex can extract all words; we ignore a few errors that arise in this procedure. This extraction results in better performance mainly because words in hash tags, to some extent, may convey sentiments of hate. They play an important role during the model-training stage.

We treat the tweet as a sequence of words with interdependence among various words contributing to its meaning. Hence we use an bidirectional LSTM based approach to capture information from both the past and future context.

Our model is a neural-network based model. First, the input tweet is passed through an embedding layer which transforms the tweet into a 128 length vector. The embedding layer learns the word embeddings from the input tweets. This is followed by two bidirectional LSTM layers containing 64 units each. This is followed by the final output layer of neurons with softmax activation, each neuron predicting a label as present in the dataset. For subtasks 1 and 2, it contains 2 neurons for predicting `OFF/NOT` and `TIN/UNT` respectively; for subtask 3 it contains 3 neurons for predicting `IND/GRP/OTH`. Between the LSTM and output layers, we add dropout with a rate of 0.5 as a regularizer. The model is trained using the Adam optimization algorithm with a learning rate of 0.0005 and using crossentropy as the loss.

We note that the dataset is highly skewed in nature. If trained on the entire training dataset without any validation, the model tends to completely overfit to the class with higher frequency as it leads to a higher accuracy score.

To overcome this problem, we took some measures. Firstly, the training data was split into two parts; one for training and one for validation comprising 70 % and 30 % of the dataset respectively. The training was stopped when two consecutive epochs increased the measured loss function value for the validation set.

Secondly, class weights were assigned to the different classes present in the data. The weights were approximately chosen to be proportional to

the inverse of the respective frequencies of the classes. Intuitively, the model now gives equal weight to the skewed classes and this penalizes tendencies to overfit to the data.

In general, we took 0.5 as the boundary between predictions of 0 and 1 — essentially rounding the predicted values. However, for subtask B, we try different values for this parameter (`thresh`) to achieve better results. Values less than `thresh` are converted to 0 while the remaining values are converted to 1.

5 Results

We have included the automatically generated tables with our results. We have also included some baselines generated by assigning the same labels for all instances. For example, "All OFF" in subtask A represents the performance of a system that labels everything as offensive. We have used this for comparison.

We have also added the relevant confusion matrices that were provided together with the results.

System	F1 (macro)	Accuracy
All NOT baseline	0.4189	0.7209
All OFF baseline	0.2182	0.2790
BiLSTM	0.4650	0.5651

Table 3: Sub-task A, garain CodaLab 528038 (Bi-Directional LSTM)

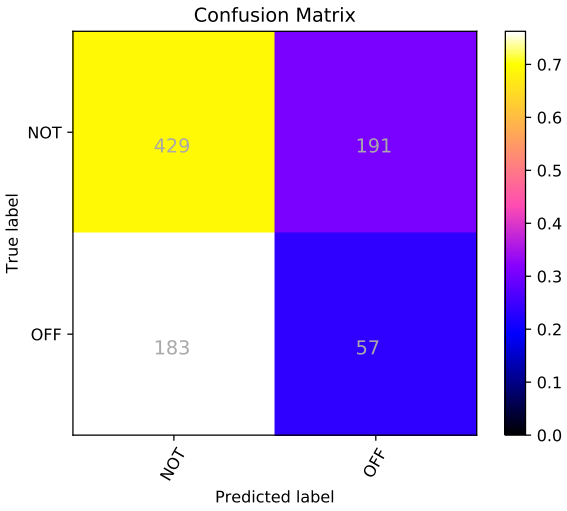

Figure 1: Sub-task A, garain CodaLab 528038 (Bi-Directional LSTM)

System	F1 (macro)	Accuracy
All TIN baseline	0.4702	0.8875
All UNT baseline	0.1011	0.1125
BiLSTM(*)	0.4702	0.8875
BiLSTM(thresh=0.50)	0.5733	0.9
BiLSTM(thresh=0.40)	**0.5796**	**0.8833**

Table 4: Sub-task B, garain CodaLab 533103 (Bi-Directional LSTM threshold = 0.40)
* - class weights not used

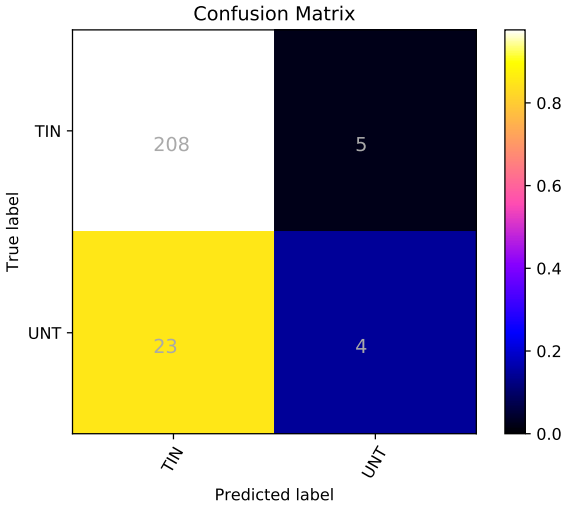

Figure 2: Sub-task B, garain CodaLab 533103 (Bi-Directional LSTM threshold = 0.4)

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
BiLSTM	0.3262	0.4601

Table 5: Sub-task C, garain CodaLab 535813 (Bi-Directional LSTM)

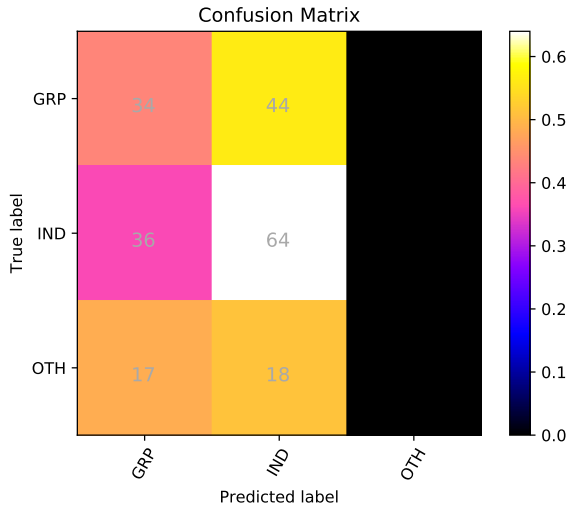

Figure 3: Sub-task C, garain CodaLab 535813 (Bi-Directional LSTM)

6 Conclusion

Here we have presented a model which performs satisfactorily in the given tasks. The model is based on a simple architecture. There is scope for improvement by including more features (like those removed in the preprocessing step) to increase performance. Another drawback of the model is that it does not use any external data other than the dataset provided which may lead to poor results based on the modest size of the data. Related domain knowledge may be exploited to obtain better results.

References

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*.

Darja Fišer, Tomaž Erjavec, and Nikola Ljubešić. 2017. Legal Framework, Dataset and Annotation Schema for Socially Unacceptable On-line Discourse Practices in Slovene. In *Proceedings of the Workshop Workshop on Abusive Language Online (ALW)*, Vancouver, Canada.

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.

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.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting Hate Speech in Social Media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 467–472.

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

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.

Huei-Po Su, Chen-Jie Huang, Hao-Tsung Chang, and Chuan-Jie Lin. 2017. Rephrasing Profanity in Chinese Text. In *Proceedings of the Workshop Workshop on Abusive Language Online (ALW)*, Vancouver, Canada.

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)*.

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.