NLP at SemEval-2019 Task 6: Detecting Offensive language using Neural Networks

Prashant Kapil[*], **Asif Ekbal**[*] **and Dipankar Das**[+]

[*]Department of Computer Science and Engineering
Indian Institute of Technology Patna, India
[+] Department of Computer Science and Engineering
Jadavpur University Kolkata, India
[*]{*prashant.pcs17, asif*}*@iitp.ac.in*
[+]*ddas@cse.jdvu.ac.in*

Abstract

In this paper we built several deep learning architectures to participate in shared task *OffensEval: Identifying and categorizing Offensive language in Social media by semEval-2019* (Zampieri et al., 2019b). The dataset was annotated with three level annotation schemes and task was to detect between offensive and not offensive, categorization and target identification in offensive contents. Deep learning models with POS information as feature were also leveraged for classification. The three best models that performed best on individual sub tasks are stacking of CNN-Bi-LSTM with Attention, BiLSTM with POS information added with word features and Bi-LSTM for third task. Our models achieved a Macro F1 score of 0.7594, 0.5378 and 0.4588 in Task(A,B,C) respectively with rank of 33^{rd}, 54^{th} and 52^{nd} out of 103, 75 and 65 submissions.

1 Introduction

Due to the exponential rise in the usage of internet user generated content in the form of blogs, posts, comments etc. have been increased manifold. Some users also using this platform to target any individual or any particular group on social media on the basis of certain attributes, sharing different views. Many studies have been conducted on offensive language, hate speech, cyberbullying, profanity, aggression detection. These contents are major concern for governments, so robust computational systems need to be developed to tackle these posts to maintain social harmony. This paper is organized as follows. Related work have been discussed in section 2, Methodology have been described in Section 3 followed by data sets and other settings used to solve the tasks in Section 4. Results and analysis of the models is described in Section 5 with the limitation of the models in Error Analysis in section 6. Section 7 contains the conclusion and Future scope.

1.1 Problem Definition

The organizers proposed a hierarchical three level annotation model and divided into three sub tasks.
Task A: This task consist of classifying between *offensive* and *not offensive* comments
Task B: The Offensive language was further needs to be classified into *Targeted(TIN)* and *UnTargeted(UNT)*.
Task C: The targeted offensive needs to be further classified into *Individual(IND)*, *Group(GRP)* and *Other(OTH)*.

2 Related Work

(Nockleby, 2000) defined hate speech as any communication that demean any person or any group on the basis of race, color, gender, ethnicity, sexual orientation, and nationality. (Kowalski et al., 2014) defined cyber aggression as using digital media to intentionally harm another person. (Schmidt and Wiegand, 2017) presents a survey on the existing research in this field and different set of features used in machine learning and Deep learning were discussed. (Silva et al., 2016) proposed and validated sentence structure to detect hate speech and also used this to construct hate speech datasets. They also provided the characteristics study to identify the main targets of hate speech in Twitter and Whisper. They designed two rules i.e *I*<intensity><user intent><hate Target> and <one word> people ex:"black people","maxican" people. (Waseem, 2016) examined the performance of classification based on training performed on amateur and expert annotations. (Ross et al., 2017) concluded that hate speech requires significantly better definitions and

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 587–592
Minneapolis, Minnesota, USA, June 6–7, 2019.

guidelines. (Sood et al., 2012) detected profanity by identifying offensive words using list based methods and incorporated edit distance to find similar obscene words. (Davidson et al., 2017) observed that seperating offensive and hate speech is very challenging task. *nigga*, *hoe* , *bitch*, *fag* are very offensive in nature but can be used in different manner. They reported Logistic Regression as their best classifier in detecting approx. 25K Tweets by using N-grams weighted by TF-IDF, POS n-grams and using sentiment score as their features.(Samghabadi et al., 2017) used surface level features like word n-grams and char n-grams, LIWC and SentiWordNet to get the sentiment score as well as some domain related features. (Malmasi and Zampieri, 2017) used char n-grams, word n-grams and word skip grams to get accuracy of 78% on Data set of 14509 tweets classified into 3 classes. (Waseem et al., 2017) tried to capture similarities between different sub tasks. They proposed a typology to differentiate language on the basis of individual or group attack or if the content is explicit or implicit. (Gambäck and Sikdar, 2017) used random vector, word vectors and also concatenated word based CNN and character based CNN to classify 6909 tweets into 4 classes. (Xu et al., 2012) used LDA to find out relevant and useful sentiment in bullying texts. (Zhang et al., 2018) proposed a CNN-GRU based structure which outperformed 6 out of 7 datasets by at most 13 F1 points. They have used surface level features, linguistic features, sentiment features as well as number of misspellings , percentage of capitalisation for SVM. (Aroyehun and Gelbukh, 2018) implemented several neural networks and also found that char n-grams is more superior than word n-grams in NBSVM. They also used data augmentation, pseudo labeling and sentiment score as feature. (Kumar et al., 2018) discuss the task of developing a classifier to discriminate *Overtly*,*Covertly* and *Non Aggressive* text using 15000 annotated social media data in both English and Hindi(in Roman and Devanagri script) as part of TRAC-1. (Badjatiya et al., 2017) experimented with several deep neural architecture and found that it outperformed state of the art word/char n-grams.(Djuric et al., 2015) proposed paragraph to vector for modelling of comments. (Gao and Huang, 2017) discusses the Bi-LSTM with attention mechanism with learning components context improved the classifier performance.

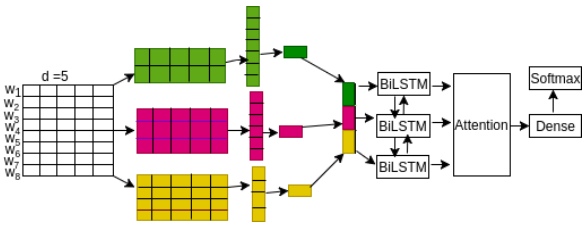

Figure 1: **Sub Task A:CNN-BiLSTM-Attention**

(Founta et al., 2018) studied different forms of abusive behaviour and made public annotated corpus of 80K Tweets categorized into 8 labels like Hate, aggressive, cyber bullying, normal , Spam.

3 Methodology

3.1 Task A:CNN-BiLSTM-Attention

In this model first we converted all the words to their unique index. Then all the unique index in the sentences were mapped to their real valued vectors of Dimensions 100 using Glove by (Pennington et al., 2014) from Embedding Matrix. Convolution layers is used to extract useful information by convolving i words at a time using learnable kernel of size i*h where i = [2,3,4] and h is of size equal to the dimensions. The element wise dot product is performed to get the feature map f_1. N numbers of filters are used to get feature map = [f_1,f_2...f_n]. Pooling reduces the size of representation by selecting max value from each feature map which is then passed to the BiLSTM layer with 100 hidden units. The sentence level representation is then passed to activation layers to capture the important keywords informations. This vector representation is then passed to softmax classification to get the probability values of each class.

Attention It tries to make RNN better by letting the network to know the weight of important keywords. It produces state of the art results on several NLP tasks. We used the approach followed by (Ding et al., 2018) for sentence level attention which follows the following equation.

$$e_t = tanh(Wh_t + b) \quad (1)$$

$$\alpha_t = softmax(e_t) \quad (2)$$

$$output = \sum_{t=1}^{t=n} \alpha_t h_t \quad (3)$$

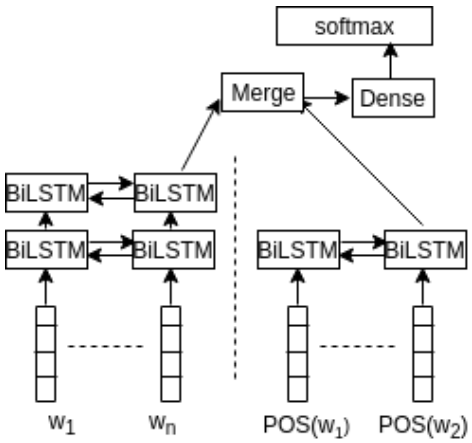

Figure 2: **Sub Task B:BiLSTM(Word+POS)**

3.2 Task B:BiLSTM(Word+POS)

In this model two layers of BiLSTM were used with hidden nodes of 100 where the sentences were being represented by Glove embedding. BiLSTM uses 2 LSTM that is useful for keeping both the past and future information. The input sequence (i_1,i_2,...i_n) is converted to (h_i^1,h_i^2...h_i^n) taking into account each words. Each word was tagged with its POS Tag and embedding for each Tag was calculated. Each sequence was then converted to their POS Tag real valued vector of Dimensions of 20 using embedding matrix. The input sequence is then passed to BiLSTM layer with hidden nodes of 100. The outputs of both the channels were concatenated and passed to the Fully connected layer followed by softmax Classification.

3.3 Task C: BiLSTM

We used BiLSTM using 100 dimensions to represent sequences by fixing the maximum length to 40 . Post padding with 0 was used for shorter sequences as it helps in preserving the information at the borders. After getting desired hidden representation from 2 layers it is passed to the Fully Connected layers followed by softmax Classifier for getting probability distribution among classes.

4 Data Sets

The Datasets provided by organisers (Zampieri et al., 2019a) were three level annotated social media text. The task was divided into three parts,description of their data sets is in Table**1**, Table **2** and Table **3** .

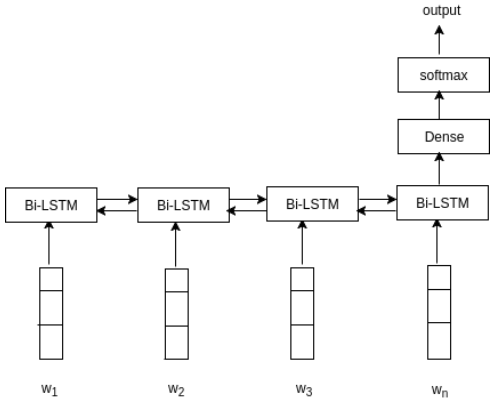

Figure 3: **Sub Task C:BiLSTM**

Class	#Training	#Test
Offensive	4400	240
Not Offensive	8840	820

Table 1: **Data set:Task A**

Class	#Training	#Test
Off_Targeted	3876	213
Off_Untargeted	524	27

Table 2: **Data set: Task B**

Class	#Training	#Test
Off_Tar_IND	2407	100
Off_Tar_GRP	1074	78
Off_Tar_OTH	395	35

Table 3: **Data set:Task C**

4.1 Parameter Tuning,Word embedding and evaluation Metrics

We use Keras with Tensorflow as backend,Scikit-learn library for implementation. For every dataset we use 80:20 for 80% to use in Training and using grid search to learn batch size and epochs. Experiments were performed using stratified 5-fold cross validation to train all the classes according to their proportion and 20% of remaining data were used as testing the model. We are reporting our results on Training data provided by orgainsers by standard Precision, Recall and F-score by averaging all the cross fold results. Categorical cross entropy loss function and Adam optimiser were used for training . In the experiment we use publicly available Glove embedding by (Pennington et al., 2014). We used batch size of(16,32,64) and drop out of (0.1,0.2,0.3).

4.2 Preprocessing

As the datasets are collected from social media it contains lots of noise and inconsistencies in the form of urls, typos and abbreviations. So we start by applying light preprocessing by expanding all appostrophes containing words and then removing characters like : , & ! ? and also all the tokens were tranformed to lower case to avoid capitalized versions of same word being treated as different words. We also used dictionary to expand the misspelled words to its original form. The POS tags were obtained from NLTK.

Class	OFF	NOT	F1	Acc.
OFF	2614	1786	74.96	78.95
NOT	1006	7834		

Table 4: **Cross validation: Task A**

Class	TIN	UNT	F1	Acc.
TIN	3839	37	51.56	88.43
UNT	472	52		

Table 5: **Cross validation: Task B**

Class	IND	GRP	OTH	F1	Acc.
IND	2103	303	1	47.69	71.18
GRP	423	649	2		
OTH	208	182	5		

Table 6: **Cross validation: Task C**

Class	OFF	NOT	F1	Acc.
OFF	131	109	75.94	82.44
NOT	42	578		

Table 7: **Test Set: Task A**

Class	TIN	UNT	F1	Acc.
TIN	212	1	53.78	89.17
UNT	25	2		

Table 8: **Test Set: Task B**

5 Results and Analysis

We have reported the cross validation split accuracy and F-score in Table 4, Table 5 and Table 6 for all the three subtasks. The results for test set is also included in Table 7 .Table 8 and Table 9. For our systems we got almost comparable results for both training and test datasets. We got F-score of 75.94%, 53.78% and 45.88% in sub task A, B, C respectively. Table 10 shows the performance of our system compared with best systems.

Class	GRP	IND	OTH	F1	Acc.
GRP	48	30	0	45.88	64.32
IND	11	89	0		
OTH	15	20	0		

Table 9: **Test Set: Task C**

Task	Ours	Best
A	75.94%	82.9%
B	53.78%	75.5%
C	45.88%	66%

Table 10: **System Performance**

6 Error analysis

Error analysis was carried out to analyze the errors that we encountered in our system by quantitative analysis using Confusion matrix of our best models for each task.

6.1 Quantitative Analysis

From Table 7 it can be seen that false negative rate of offensive class is 45% where as for Not Offensive True Positive rate is 93.22% in Task 1. 42 instances of Not Offensive also got misclassified as Offensive showing evidence of challenges in classification. For Task2 Table 8 shows that TIN True positive rate is almost 100% but system fails to classify UNT class with only 0.08% true positive rate. For Task3 Table 9 shows that system completely fails to detect OTH class with false negative rate of 100%. However GRP and IND class obtained True positive rate of 61.5% and 89% respectively . The misconversion instances of GRP and IND to each other is 30 and 11.

7 Conclusion and Future Scope

In this paper we have explored the effectiveness of deep neural network for Offensive speech detection. We can conclude that fine grained analysis of offensive language detection needs careful attention. Linguistic features can also be leveraged for improvement in classifier.

References

Segun Taofeek Aroyehun and Alexander Gelbukh. 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97.

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*, pages 759–760. International World Wide Web Conferences Steering Committee.

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

Zixiang Ding, Rui Xia, Jianfei Yu, Xiang Li, and Jian Yang. 2018. Densely connected bidirectional lstm with applications to sentence classification. *arXiv preprint arXiv:1802.00889*.

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 Companion*, pages 29–30. International World Wide Web Conferences Steering Committee.

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. *arXiv preprint arXiv:1802.00393*.

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.

Lei Gao and Ruihong Huang. 2017. Detecting online hate speech using context aware models. *arXiv preprint arXiv:1710.07395*.

Robin M Kowalski, Gary W Giumetti, Amber N Schroeder, and Micah R Lattanner. 2014. Bullying in the digital age: A critical review and meta-analysis of cyberbullying research among youth. *Psychological bulletin*, 140(4):1073.

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 1–11.

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.

John T Nockleby. 2000. Hate speech. *Encyclopedia of the American constitution*, 3:1277–79.

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.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2017. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*.

Niloofar Safi Samghabadi, Suraj Maharjan, Alan Sprague, Raquel Diaz-Sprague, and Thamar Solorio. 2017. Detecting nastiness in social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 63–72.

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.

Leandro Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. In *Tenth International AAAI Conference on Web and Social Media*.

Sara Sood, Judd Antin, and Elizabeth Churchill. 2012. Profanity use in online communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1481–1490. ACM.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142.

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*.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 656–666. Association for Computational Linguistics.

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*.

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). *arXiv preprint arXiv:1903.08983*.

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.