# Pardeep at SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media Using Deep Learning

**Pardeep Singh**
School of Computer & Systems Sciences
Jawaharlal Nehru University
New Delhi-110067
pardeepsinghinfo@gmail.com

**Satish Chand**
School of Computer & Systems Sciences
Jawaharlal Nehru University
New Delhi-110067
schand20@gmail.com

## Abstract

The rise of social media has made information exchange faster and easier among the people. However, in recent times, the use of offensive language has seen an upsurge in social media. The main challenge for a service provider is to correctly identify such offensive posts and take necessary action to monitor and control their spread. In this work, we try to address this problem by using sophisticated deep learning techniques like LSTM, Bidirectional LSTM and Bidirectional GRU. Our proposed approach solves 3 different Sub-tasks provided in the SemEval-2019 task 6 which incorporates identification of offensive tweets as well as their categorization. We obtain significantly better results in the leader-board for Sub-task B and decent results for Sub-task A and Sub-task C validating the fact that the proposed models can be used for automating the offensive post-detection task in social media.

## 1 Introduction

Social media has revolutionized the way of communication among the people. It is an instant communication medium which connects people all over the world and shares their views. But, some people misuse this freedom by using the offensive language through posts or comments to defame, insult or target an individual or a group of individuals. The mainstream media have reported various cases of suicide and depression due to trolling and cyberbullying in social media. Hence it becomes worrisome for the corporates, government organizations and security agencies to either stop or mitigate this type of behavior of the users. Manually it is impossible to check the negative behavior of users due to the volume, velocity and variety of data coming from the social networks. Hence there is an utmost need to develop a system which automatically identifies and categorizes the offensive language in social networks. To tackle these issues SemEval-2019 (Zampieri et al., 2019b) aimed exactly at that need and organized a task in identifying and categorizing offensive language in social media. This task is divided into three Sub-tasks.
Sub-task A - Offensive language identification.
Sub-task B - Categorization of offense types.
Sub-task C - Offense target identification.
All the three Sub-tasks are related to each other. In Sub-task A, we have to identify whether a given set of tweets is offensive or not. It is a binary classification task based on tweet text. In Sub-task B, the main challenge is to categorize the tweets which are offensive in Sub-task A into targeted or untargeted. Sub-task C is comparatively challenging than other two Sub-tasks due to the multi-class nature. Its goal is to identify the tweets which are targeted in Sub-task B and categorized those tweets into individual, group or others.
Our approach for the SemEval-2019 task 6 (identifying and categorizing offensive language in social media) comprises of deep learning models: Bidirectional LSTM, Bidirectional GRU and standard LSTM. These are popularly used deep learning sequence models applied in many text classification tasks. We used the pre-trained word level embedding GloVe (Global Vectors for Word Representation) to get vector representations for words that appeared in tweets and used these representations as features for training the models. To check the performance of models, 10 fold cross-validation was applied on the given training data. We compared the results of the above-mentioned models with various baselines such as Logistic Regression, Support Vector Machine, Gradient Boosting and XGBoost. The baseline models are reasonably good but they have poor classification Accuracy as compared to deep learning models. This paper presents the description of our approaches and results for SemEval-2019 task 6.

*Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*, pages 727–734
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

## 2 Related Work

This section discusses some existing work related to identifying and categorizing offensive language in social media. Researchers have applied various computational methods to deal with hate speech, aggression, offensive language, racist and sexist language, and cyberbullying.

**Hate Speech:** Detection of hate speech is modeled in (Zhang et al., 2018). The authors applied CNN and GRU deep neural networks along with pre-trained Google Word2vec word embedding to detect the hate speech on Twitter. (Zhang and Luo, 2018) proposed Skip Gram Extraction CNN (SKIP-CNN) deep neural network model to identify hate speech present in social media text. It is discussed in this paper that hate speech lacks distinctive and unique features in a dataset which is hard to discover. The proposed model serves as a feature extractor for capturing the semantics of hate speech in social media.

**Aggression:** A method to detect aggression in social media is proposed in (Madisetty and Desarkar, 2018). The authors applied CNN, LSTM and Bidirectional LSTM on Facebook comment dataset. The output of these three deep learning models are used as an input to the majority based ensemble method for detection of aggression in social media. Another paper (Kumar et al., 2018) presents the system description report of shared task on identification of aggression in social media as a part of the 1st workshop on trolling, aggression and cyberbullying (TRAC1). The aggression annotated dataset of Facebook posts and comments in English and Hindi language were provided to the participants for training and validation. Six models out of the top ten best performing models were trained using LSTM, Bidirectional LSTM, CNN, and RNN deep neural networks.

**Racist and Sexist Language:** (Davidson et al., 2017) focused on classifying homophobic and racist tweets as hate speech and sexist remarks tweets as offensive. They use Logistic Regression with L2 regularization to predict the class membership. (Pitsilis et al., 2018) proposed an ensemble LSTM deep learning classifier that utilizes the user behavior metric to show each user viewpoint towards racism and sexism captured by their tweeting history.

**Cyberbullying:** (Dadvar et al., 2013) studied about the Cyberbullying detection. They combine individual comments, user characteristics and user profile information for training the Support Vector Machine classifier. It is also reported that the addition of user history with text features improves cyberbullying detection accuracy. (Rafiq et al., 2018) proposed a multi-stage cyberbullying detection mechanism by two novel components. First is dynamic priority scheduler which drastically reduces the classification time, and second is incremental classification method which is highly responsive regarding time to raise alerts.

Until now there have been many publications and studies on offensive language, aggression and hate speech in social media. Examples include (Wiegand et al., 2018), (ElSherief et al., 2018) and (Fortuna and Nunes, 2018). All these methods have some pros and cons associated with them. Therefore this paper proposed the idea of using deep learning sequence models for better accuracy in results for SemEval-2019 task 6.

## 3 Methodology and Data

In this section, we first describe the dataset used in the competition and then we explain the description of approaches used for solving the problem.

### 3.1 Dataset Used

The dataset provided by the task organizers is OLID (Offensive Language Identification). The details of data and annotation are available in (Zampieri et al., 2019a). For Sub-task A, this dataset contains tweets labeled into the following two categories: offensive (OFF) and not offensive (NOT). For Sub-task B, tweets are labeled into the following two categories: targeted input (TIN) and untargeted (UNT). For Sub-task C, the given tweets are classified into the following three categories: group (GP), individual (IND) and others (OTH). Out of 13,240 training samples of Sub-task A, 4404 samples have been allocated to Sub-task B and 3,877 samples have been allocated to Sub-task C. All the tweets are in English language. The statistics of the dataset and some instances of tweets with their labels are shown in Table 1 and Table 2.

| | Training Set samples | Testing Set samples |
|---|---|---|
| Sub-task A | 13240 | 860 |
| Sub-task B | 4404 | 240 |
| Sub-task C | 3877 | 213 |

**Table 1:** Statistics of the offensive dataset

| Tweet | Sub-task A | Sub-task B | Sub-task C |
|---|---|---|---|
| Its not my fault you support gun control. | NOT | - | - |
| Someone should'veTaken" this piece of shit to a volcano. | OFF | UNT | - |
| you are a lying corrupt traitor!!! Nobody wants to hear anymore of your lies!!! #DeepStateCorruption. | OFF | TIN | IND |
| Kind of like when conservatives wanna associate everyone to their left as communist antifa members? | OFF | TIN | GRP |
| why report this garbage. We don't give a crap. | OFF | TIN | OTH |

**Table 2:** Some tweets from the training dataset with their labels.

### 3.2 Methodology

Here we discuss our proposed approach in details. Our initial approach was to check with standard machine learning algorithms like Logistic Regression (Hosmer Jr et al., 2013), Random Forest (Xu et al., 2012), Support Vector Machines (Chang and Lin, 2011), XGBoost (Chen and Guestrin, 2016) and Gradient Boosting (Natekin and Knoll, 2013). We use TF-IDF vectorization for vectorizing our text and then apply the above-mentioned algorithms for the model development. Performance of these algorithms were not quite acceptable as it gave low Accuracy in results. To overcome above mentioned issues, we use deep learning algorithms for classifying the text. First we convert the text into vector representations with the help of GloVe (Pennington et al., 2014) word level embeddings and then use these representations as an input to the deep learning models described in the subsequent sections for classification tasks.

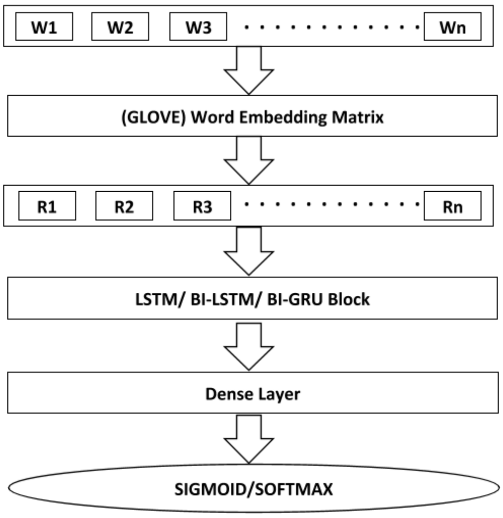

**Figure 1:** Architecture of the proposed deep learning models.

The multi-layered architecture of our approach presented in Figure 1. It comprised of various components in the form of layers. Since the data is in the form of text and the first step is to vectorize the text. To achieve this, we first make the tokens of the text W1, W2, W3,..., Wn and apply pre-trained GloVe word embeddings to get vector representations R1, R2, R3,..., Rn from it. Next layer can be LSTM, Bidirectional LSTM or Bidirectional GRU Block described in the next subsections. To overcome the problem of overfitting, we add a small amount of dropout. Finally, we use a Dense layer and Softmax/Sigmoid layer to get the output of the models.

#### 3.2.1 Long Short Term Memory (LSTM)

We first try LSTM (Hochreiter and Schmidhuber, 1997) which have been used successfully in many text classification tasks (Madisetty and Desarkar, 2018). LSTM is special kind of RNN which captures the long contexts and long-range dependencies very efficiently in the sentences and takes care of the vanishing gradient problem of RNN (Lipton et al., 2015) with the help of carefully regulated structures named gates. The main components of the LSTM model are input gate, forget gate, output gate and candidate memory state. All these gates are single layered neural networks with the Sigmoid activation function except candidate memory state which uses tanh as the activation function.

#### 3.2.2 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (Tjandra et al., 2016) is an improvisation over LSTM. They also take care of the vanishing gradient problem of the RNN and tries to capture long-range connections better but with a less number of gates than LSTM. This leads to a less amount of parameters for the model which enables a faster and efficient model development in comparison to the LSTM based model.

The main components of GRU are reset gate, update gate and current memory content. Similar to LSTM, both reset gate and update gate are single layered neural networks with the Sigmoid activation function except current memory content which use tanh as the activation function. The basic function of the reset gate is to determine how much of the past information to be lost whereas the update gate decides how much of the information the model should pass to the next states.

### 3.2.3 Bidirectional LSTM and GRU

Both LSTM and GRU uses sequential information of the textual data for the processing and capture much longer range dependencies. But, the catch is that they use the sequence of only one direction while the Bidirectional version of the same considers a reverse copy of the provided input. In certain problems, this reversal helps to a better feature understanding and improved model performance.

In our work, we mainly use standard LSTM and Bidirectional version of both LSTM and GRU for the model developments. The detailed experimental setup is described in the next section.

## 4 Experimental Setting

For implementing the models, we use Keras (Ketkar, 2017) and Scikit-learn (Pedregosa et al., 2011) python framework libraries. The experimental details and model configuration are shown in Table 3. For the effectiveness of models, we add a small proportion of dropout. For GRU model, we specify the number of Recurrent Units. In terms of training, we use categorical cross Entropy as a loss function with ADAM as the optimization function. All the models are tested using 10 fold cross-validation.

| **Model Configuration** | **Value** |
|---|---|
| sentences_length | 32 |
| batch_size | 64 |
| recurrent_units (for GRU) | 64 |
| dense_size | 16 |
| dropout_rate | 0.5 |
| number of epochs | 300 |

**Table 3:** Configuration of the proposed models.

### 4.1 Impact of Batch Size on Model Performance

We checked our proposed approach with three different batch_sizes 64, 128 and 256 to check its impact on model performance. It is found experimentally that batch_size 64 provides optimal results.

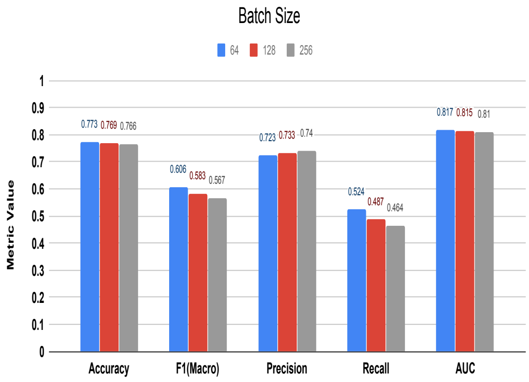

**Figure 2:** Performance comparisons with different batch Sizes.

**Performance metrics**: The official evaluation metric for all the three Sub-tasks are the macro-averaged F1 score. For additional analysis, we use the Accuracy, Precision, Recall and ROC-AUC.

## 5 Results and Discussions

This section contains the detailed experimental results that we performed on the proposed models including the baselines. It is quite familiar that multiple baselines approaches are helpful for comparing the performance of models on validation sets. To observe this, we apply various computational models on the training data released for Sub-task A so that we figure out which models give better results on the training data. Table 4 presents each model results in terms of Accuracy, F1(Macro), Precision, Recall and Roc-Auc score. It is evident from this Table that the deep learning models like LSTM, Bidirectional LSTM and Bidirectional GRU with GloVe word embeddings outperformed TF-IDF based machine learning algorithms. The LSTM model provides better results in terms of Accuracy among all the models. Bidirectional LSTM provides better results in terms of F1 macro and the Random Forest with TF-IDF gives better results in terms of Precision. The Bidirectional GRU provides better results for Recall matrix. The standard LSTM and Bidirectional LSTM performs equally good in terms of ROC-AUC.

| Classifier | Accuracy | F1(Macro) | Precision | Recall | ROC-AUC |
|---|---|---|---|---|---|
| Logistic Regression + TF-IDF | 0.7610 | 0.5563 | 0.5070 | 0.4463 | 0.6832 |
| Random Forest + TF-IDF | 0.7567 | 0.5189 | **0.7718** | 0.3908 | 0.6662 |
| SVM with linear Kernel + TF-IDF | 0.7658 | 0.5579 | 0.7613 | 0.4403 | 0.6853 |
| Xgboost + TF-IDF | 0.7323 | 0.5248 | 0.6493 | 0.4403 | 0.6601 |
| Gradient Boosting + TF-IDF | 0.7525 | 0.5750 | 0.6785 | 0.4988 | 0.6897 |
| BI-LSTM + GloVe | 0.7686 | **0.6089** | 0.7026 | 0.5459 | **0.8100** |
| BI-GRU + GloVe | 0.7524 | 0.6021 | 0.6501 | **0.5672** | 0.7963 |
| LSTM + GloVe | **0.7695** | 0.5942 | 0.7191 | 0.5081 | **0.8100** |

**Table 4:** Results of the proposed deep learning approaches including baselines on the Sub-task A training data using 10-fold cross validation

### 5.1 Results for Sub-task A

The official results of our proposed models on the test set for Sub-task A is shown in Table 5. As it is evident from these results that Bidirectional GRU performed better than other two deep learning models with F1 Score of 0.69. To analyze the correct label of a tweet, we also show the confusion matrix which shows correct class predictions along diagonal lines. Our team ranked 74 out of 104 participating teams.

| System | F1 (macro) | Accuracy |
|---|---|---|
| All NOT baseline | 0.4189 | 0.7209 |
| All OFF baseline | 0.2182 | 0.2790 |
| BI-LSTM | 0.6617 | 0.7535 |
| **BI-GRU** | **0.6992** | **0.7744** |
| LSTM | 0.6785 | 0.7477 |

**Table 5:** Results for Sub-task A (Binary Classification)

### 5.2 Results for Sub-task B

As comparison to Sub-task A, the official results for Sub-task B shown in Table 6 are significantly better. Our team ranked 7 out of 76 participating teams with F1 Score of 0.69. Again the Bidirectional GRU outperforms both LSTM and Bidirectional LSTM deep learning models in terms of F1 and Accuracy.

| System | F1 (macro) | Accuracy |
|---|---|---|
| All TIN baseline | 0.4702 | 0.8875 |
| All UNT baseline | 0.1011 | 0.1125 |
| BI-LSTM | 0.6511 | 0.8958 |
| **BI-GRU** | **0.6997** | **0.9** |
| LSTM | 0.6455 | 0.8917 |

**Table 6:** Results for Sub-task B (Binary classification)

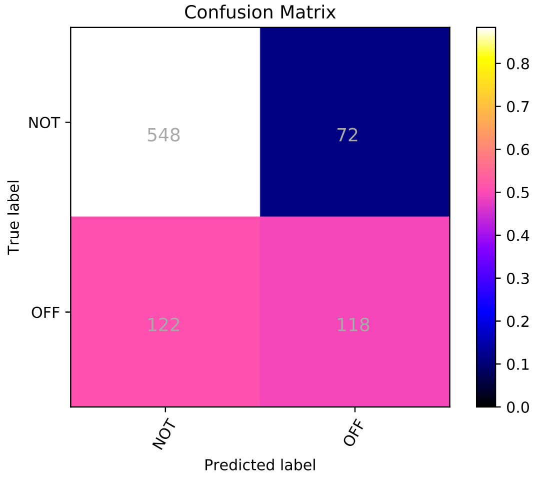

**Figure 3:** Confusion Matrix shows results for Sub-task A using Bidirectional GRU

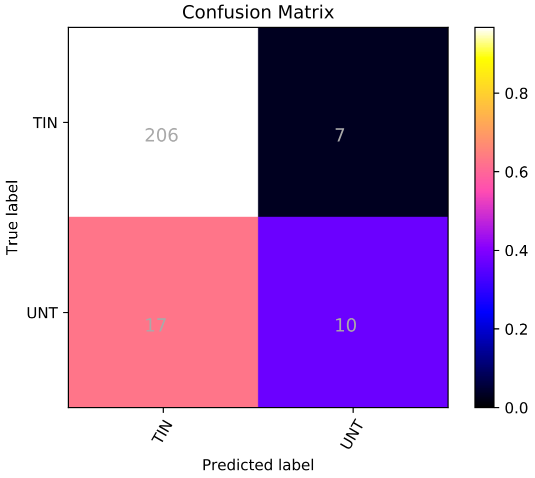

**Figure 4:** Confusion Matrix shows results for Sub-task B using Bidirectional GRU

### 5.3 Results for Sub-task C

Table 7 presents our results on the test set for Sub-task C. For this multi-class classification challenge, the results are lower as compared to other two Sub-tasks. All the participating teams had a lower performance with highest F1 score of 0.66, demonstrating the difficulty of the Sub-task. Our team ranked 41 out of 65 participating teams and Bidirectional LSTM give better results with F1 score of 0.49.

| 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 |
| **BI-LSTM** | **0.4903** | **0.6056** |
| BI-GRU | 0.4635 | 0.5775 |
| LSTM | 0.4810 | 0.6197 |

**Table 7:** Results for Sub-task C (Multi-Class Classification)

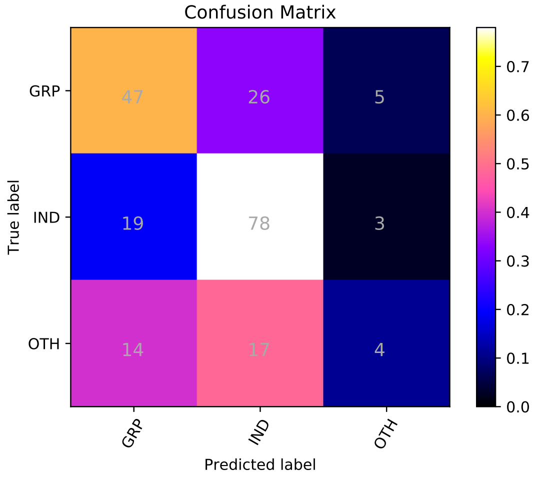

**Figure 5:** Confusion Matrix shows results for Sub-task C using Bidirectional LSTM

### 5.4 Class Label results

Besides the combined results of our proposed models on the test set for three Sub-tasks, we also present per class results in Tables 8, 9 and 10. The results in these tables show how well our models performed on each class label.

Table 8 shows each class label results for Sub-task A which comprises of two classes offensive (OFF) and not offensive (NOT). Bidirectional GRU performed better on both classes with F1 score of 0.54 and 0.84 respectively validating the fact that offensive class are relatively difficult to classify.

Table 9 shows each class label results for Sub-task B which comprises of two classes targeted input (TIN) and untargeted (UNT). Bidirectional GRU performed better on both classes with F1 score of 0.94 and 0.45 respectively which shows that untargeted class are much harder to classify.

Table 10 shows each class label results for Sub-task C which is a multi-class classification challenge having three classes individual (IND), group (GRP) and others (OTH). LSTM provides better results with F1 score of 0.72 and 0.63 for both (IND) and (GRP) classes while Bidirectional LSTM performed better on (OTH) class with F1 score of 0.17, justifies that (OTH) class is relatively harder to classify.

## 6 Conclusion

In this paper, we address the challenge of identification of offensive tweets as well as their categorization. Our proposed approach comprises of three deep learning based techniques for efficient classification of offensive posts in social media. In this work, we show that applying word embedding over social media text followed by the application of a sequence to sequence models like LSTM, Bidirectional LSTM and Bidirectional GRU leads to a better classification of the text. This proposed approach can also be incorporated in an end-to-end framework. Overall, our approach provides an efficient way of text classification in social media. For future work, we want to include character-based embeddings along with pre-trained word level embeddings for better representation of text. Also, the addition of attention layer to the deep networks sometimes increases performance even further.

### Acknowledgements

The aforementioned work is funded by the University Grants Commission (UGC), New Delhi under Senior Research Fellowship (SRF). We would also like to acknowledge Dr. Srijith P.K. and Uddipta Bhattacharjee from the Department Of Computer Science IIT Hyderabad for their support and valuable suggestion.

|  | OFF |  |  | NOT |  |  |
|---|---|---|---|---|---|---|
|  | P | R | F1 | P | R | F1 |
| BI-LSTM | 0.5814 | 0.4167 | 0.4854 | 0.7965 | **0.8839** | 0.8379 |
| BI-GRU | **0.6211** | 0.4917 | **0.5488** | **0.8179** | **0.8839** | **0.8496** |
| LSTM | 0.5520 | **0.5083** | 0.5293 | 0.8153 | 0.8403 | 0.8276 |

**Table 8:** Shows per-class performance of our proposed models for Sub-task A.

|  | TIN |  |  | UNT |  |  |
|---|---|---|---|---|---|---|
|  | P | R | F1 | P | R | F1 |
| BI-LSTM | 0.9123 | 0.9765 | 0.9433 | 0.5833 | 0.2593 | 0.3590 |
| BI-GRU | **0.9238** | 0.9671 | **0.9450** | **0.5882** | **0.3704** | **0.4545** |
| LSTM | 0.9119 | **0.9718** | 0.9409 | 0.5385 | 0.2593 | 0.3500 |

**Table 9:** Shows per-class performance of our proposed models for Sub-task B.

|  | IND |  |  | GRP |  |  | OTH |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | P | R | F1 | P | R | F1 | P | R | F1 |
| BI-LSTM | 0.6446 | 0.7800 | 0.7059 | 0.5875 | 0.6026 | 0.5949 | **0.3333** | **0.1143** | **0.1702** |
| BI-GRU | 0.6389 | 0.6900 | 0.6635 | 0.5667 | **0.6538** | 0.6071 | 0.2000 | 0.0857 | 0.1200 |
| LSTM | **0.6508** | **0.8200** | **0.7257** | **0.6575** | 0.6154 | **0.6358** | 0.1429 | 0.0571 | 0.0816 |

**Table 10:** Shows per-class performance of our proposed models for Sub-task C.

## References

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving cyberbullying detection with user context. In *Advances in Information Retrieval*, pages 693–696. Springer.

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479.

David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. 2013. *Applied logistic regression*, volume 398. John Wiley & Sons.

Nikhil Ketkar. 2017. Introduction to keras. In *Deep Learning with Python*, pages 97–111. Springer.

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.

Zachary C Lipton, John Berkowitz, and Charles Elkan. 2015. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

Sreekanth Madisetty and Maunendra Sankar Desarkar. 2018. Aggression detection in social media using deep neural networks. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 120–127.

Alexey Natekin and Alois Knoll. 2013. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier

Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

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.

Georgios K Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Detecting offensive language in tweets using deep learning. *arXiv preprint arXiv:1801.04433*.

Rahat Ibn Rafiq, Homa Hosseinmardi, Richard Han, Qin Lv, and Shivakant Mishra. 2018. Scalable and timely detection of cyberbullying in online social networks. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pages 1738–1747. ACM.

Andros Tjandra, Sakriani Sakti, Ruli Manurung, Mirna Adriani, and Satoshi Nakamura. 2016. Gated recurrent neural tensor network. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 448–455. IEEE.

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*.

Baoxun Xu, Joshua Zhexue Huang, Graham Williams, Qiang Wang, and Yunming Ye. 2012. Classifying very high-dimensional data with random forests built from small subspaces. *International Journal of Data Warehousing and Mining (IJDWM)*, 8(2):44–63.

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 and Lei Luo. 2018. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, (Preprint):1–21.

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.