TECHSSN at SemEval-2020 Task 12: Offensive Language Detection Using BERT Embeddings

Rajalakshmi Sivanaiah, Angel Deborah S, S Milton Rajendram, Mirnalinee T T
Department of Computer Science and Engineering
SSN College of Engineering
Chennai 603 110, Tamil Nadu, India
rajalakshmis@ssn.edu.in, angeldeborahs@ssn.edu.in
miltonrs@ssn.edu.in, mirnalineett@ssn.edu.in

Abstract

This paper describes the work of identifying the presence of offensive language in social media posts and categorizing a post as targeted to a particular person or not. The work developed by team TECHSSN for solving the Multilingual Offensive Language Identification in Social Media (Task 12) in SemEval-2020 involves the use of deep learning models with BERT embeddings. The dataset is preprocessed and given to a Bidirectional Encoder Representations from Transformers (BERT) model with pretrained weight vectors. The model is retrained and the weights are learned for the offensive language dataset. We have developed a system with the English language dataset. The results are better when compared to the model we developed in SemEval-2019 Task6.

1 Introduction

The usage of offensive or hate words is increasing these days, largely in online communication. People find it easier to express their opinions and thoughts in online sources rather than do it personally. The anonymity provided by the online environment encourages many people to express their views in aggressively. The information spread rate is extremely fast in online social media. People, without checking the validity of the information they receive, spread it to others.

The text content posted in messages, websites, social media and blogs are highly unstructured, informal, often misspelt, and use shorthanded notations, emojis and emoticons. Getting the meaning in the natural text is a complicated task. There is ongoing research in the field of offensive language or hate speech detection, yet it remains only a goal to obtain a full-fledged model. We have participated in the offensive language task conducted in SemEval-2019 by Zampieri et al. (2019a) with various machine learning and deep learning models in Rajalakshmi et al. (2019a). For SemEval-2020 by Zampieri et al. (2020), we have developed and tested deep learning models with different word embeddings. We have used GloVe (Pennington et al., 2014), Word2Vec (Mikolov et al., 2015) and BERT embedding (Devlin et al., 2018) pretrained models to identify the presence of offensiveness in tweets. We have participated in subtask A (OFF/NOT) of classifying whether a tweet is offensive or non-offensive and subtask B (TIN/UNT) of classifying whether an offensive tweet is targeted to anyone or untargeted.

The rest of the paper is organized as follows. Section 2 surveys the related work in this field. Section 3 describes the methodology used to solve the task. Results are discussed in section 4 and conclusion in section 5.

2 Related Work

Recently, we have seen great strides in the research on profanity speech detection in social media, which includes hate speech detection, offensive language identification, and abusive language detection. Several workshops such as GermEval, SemEval, HatEval, and TRAC gain attention of the researchers in this field. Research in hate speech includes work done by Basile et al. (2019), Fortuna and Nunes (2018), Malmasi and Zampieri (2017). Difference between profanity and hate speech, and the challenges involved are discussed in Malmasi and Zampieri (2018). An offensive language detection system is desribed out by

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2190–2196
Barcelona, Spain (Online), December 12, 2020.

Davidson et al. (2017) and Mandl et al. (2019). Most of the work use the machine learning and deep learning techniques.

The work done by Wu et al. (2019) uses BERT uncased model with an F1 score of 0.8057 for task A and 0.50 for task B. Pavlopoulos et al. (2019) uses perspective API and BERT cased and uncased models to detect the offensive language with an F1 score of 0.7933 for task A and 0.6817 for task B. SemEval-2019 Task6 report of Zampieri et al. (2019b) says that machine learning algorithms such as Support Vector Machine (SVM), logistic regression, and Artificial Neural Network (ANN), and deep learning techniques such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Bidirectional Long Short-Term Memory network (BiLSTM), Embeddings from Language Models (ELMo), Bidirectional Encoder Representations from Transformers (BERT), Gated Recurrent Unit (GRU) and ensemble techniques have been employed for offensive language detection. Out of the 115 participant teams, 70% of the teams used deep learning techniques. In that, 20% teams used ensemble techniques, 11% used CNN, 13% used LSTM & BiLSTM, 10% used RNN & GRU, 8% used BERT and the remaining used other DL methods. Ensemble and BERT models gave better results for the top teams when compared to the other techniques.

3 System Methodology

We have participated in SemEval-2019 Task 6 to identify and categorize English tweets using machine learning and deep learning models Rajalakshmi et al. (2019a), where it was found that 1D-CNN with GloVe embeddings and 2D-CNN with Word2Vec embeddings have performed better compared to other machine learning and deep learning algorithms. For SemEval-2020 Task 12, in addition to these algorithms, we also used deep learning model with BERT embeddings. The proposed system comprises of the following modules:

- Dataset preparation
- Preprocessing
- Train the model using deep learning techniques
- Prediction performance of the model

3.1 Dataset Preparation

The Semi-Supervised Offensive Language Identification Datatset (SOLID) used for developing the system comprises 9,089,140 instances in training dataset for subtask A and 3,887 instances in test dataset. It has 188,974 offensive instances in training dataset for subtask B and 1,422 instances in test dataset listed by Rosenthal et al. (2020). The break up of the dataset and its classes are shown in Table 1. Since the available infrastructure in our lab does not support working with the entire dataset for subtask A, we have used 1,000,000 instances to train the model. Each entry in the dataset consists of the features `Id, Tweet, Avg_conf and Conf_std`. `Avg_conf` is the average of the confidences predicted by several supervised models for a specific instance to belong to the positive class for that subtask. The class labels are OFF (offensive) and NOT (not offensive) for subtask A, while TIN (targeted insult and threat) and UNT (untargeted) for subtask B. `Conf_std` is the standard deviation of confidences for a particular instance. We have taken the threshold to be 0.5 for `Avg_conf` and the values greater than the threshold are taken as positive examples and the rest as negative examples. The dataset is prepared in the format `Id, Tweet, Label` suitable to train the model.

Task	Label	Train	Test
A	OFF	1,448,861	1,080
	NOT	7,640,279	2,807
B	TIN	149,550	850
	UNT	39,424	572

Table 1: Data distribution of SOLID dataset for various class labels

3.2 Preprocessing

Unstructured tweet data contains a lot of irregularities which will affect the accuracy of the model. Therefore, it is important to preprocess the data before using it to build the model. Data preprocessing is an important step for increasing the performance of the model. The data is preprocessed by removing the irregularities, smoothening and normalizing the dataset. We have used NLTK (Bird et al., 2009) and Spacy toolkits (Honnibal and Montani, 2017) to preprocess the dataset. The preprocessing steps that we have developed in Rajalakshmi et al. (2019a) are listed below. Step e. is omitted for subtask B, since stopwords are significant only for target identification.

a. URL removal
b. Emojis and emoticons annotation
c. Uppercase to lowercase conversion
d. Contractions expansion
e. Stopwords removal
f. Special characters removal
g. Accented characters removal
h. Lengthened words reduction
i. Text lemmatization
j. Extra whitespace removal

We preprocessed the data with the steps outlined above and built the model using plain text. Furthermore, we can analyze the importance of accented characters, special characters and fully uppercase words and how they affect the performance of the system.

3.3 Model Building

Various deep learning techniques with different word embeddings are applied on the SOLID dataset and their performances are analyzed. Our work in SemEval-2019 task 6 showed that 2D-CNN model with Word2Vec embeddings and 1D-CNN model with GloVe embeddings performed better than all other machine learning and deep learning algorithms with different word embeddings. For the present task, we have used LSTM and BERT models in addition to those algorithms.

3.3.1 2D-CNN with Word2Vec Learned Embeddings

We have used 2D-Convolutional Neural Network (CNN) model with Google's Word2Vec pretrained weights as in Rajalakshmi et al. (2019b). The model is then retrained to relearn the weights for OffensEval2020 SOLID dataset. The structure of the model comprises of the following layers.

1. Input layer
2. Embedding layer
3. Convolutional layer with kernel size 2, 3 and 4
4. Pooling layers for CNN layers
5. Fully connected dense layer
6. Output layer

Embedding layer is used to relearn the weights of embedding matrix. Kernel filters are used to process bigrams, trigrams and fourgrams. Max pooling layer is used to scale down the output vectors to dense feature vectors that are concatenated and flattened in fully connected layer. Output layer consists of 2 units for OFF/NOT or UNT/TIN. The word-grams are concatenated and computed in parallel to extract the possible information from the vectors. This enables the classifier to understand the relationship between the words. The parameters for the model are set as follows: sequence length of the model is 43, learning rate is set as 0.001 and dropout is set as 0.5. Softmax activation function is used for output layer and Relu activation function in other layers.

3.3.2 1D-CNN with GloVe

Conventional CNN with one dimensional layer is used with GloVe embeddings with 1 million word vectors of 200 dimensions from twitter data. The embedding layers are used to extract the skip-grams. Convolutional 1D layers use kernel filters of size 2, 3 and 4. Dropout value is set as 0.2 and 100 filters are also used. Maxpooling 1D layers are used to select the bigram, trigram and fourgram branches and they are merged for further processing. Softmax function is used in output layer and Relu function is used in all other layers. This model has fewer trainable parameters and takes less time to train.

3.3.3 BiLSTM

Recurrent Neural Network (RNN) is especially designed to work with sequential data. Long Short-Term Memory networks (LSTM) (Hochreiter and Schmidhuber, 1997), an extension of RNN, are connected in a special way to avoid vanishing and exploding gradient issues. Bidirectional LSTMs can capture information about the past and future states simultaneously. We have used 2 LSTM layers for bidirection with 150 units and the inputs are trained with a batch size of 128 and dropout value of 0.2. Sigmoid function is used as the activation function in output layer and Adam algorithm is used for optimization.

3.3.4 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a deep bidirectional network built using transformers, which is pre-trained to detect a masked word in the given context sentence described by Devlin et al. (2018). It can also be used for text classification and semantic relation extraction. We have used the publicly available BERT-Base, Multilingual cased pre-trained model for 104 languages that includes English, Tamil, Telugu, Hindi, Spanish, Arabic, Turkish, Urdu, Danish, Chinese, French and, Greek etc. This model has 12 transformer layers, 768 hidden layers and 12 heads with 110M parameters. This model is well suited for any of the given languages for task 12 given by Zampieri et al. (2020).

GloVe and Word2Vec embeddings are context-free, while BERT embeddings use contextual representation for word embeddings. Context-free representation gives a single word embedding for each word irrespective of its prefix or suffix. Hence the word “bank” in “bank deposit” and “river bank” has same representation. In contextual representation the word “bank” has different representations based on the context of its nearby words; the contextual representation is considered in both forward and backward directions. Since BERT uses contextual knowledge in decision making, it will provide better interpretation than the context-free GloVe and Word2Vec embeddings as shown in the results.

We have used the CoLA (The Corpus of Linguistic Acceptability) dataprocessor given in Warstadt et al. (2018) that is mainly used for single sentence classification task. The sequence length is set to 128, and the batch size to 32. Adam optimizer is used and the learning rate is set as 0.2 to minimize the training time since the data is huge. The tweet sequence is tokenized and converted into features. The model is initialized with pretrained weight vectors and retrained to learn the input dataset features.

4 Results and Discussion

Various deep learning models with different word embeddings are used to detect the presence of offensiveness in the given tweet and to identify whether the tweet is targeted or not. Table 2 shows the models used to classify the given tweet into offensive or non-offensive category in subtask A. 1D-CNN model is trained with GloVe pretrained embeddings, 2D-CNN and BiLSTM models with Word2Vec embeddings. Deep network model with BERT embeddings achieves better F1 score when compared to other models.

Model used	F1 Macro
1D-CNN	0.732
2D-CNN	0.7451
BiLSTM	0.712
BERT	**0.86551**

Table 2: Results for models used in subtask A

Table 3 shows the results for classifying the tweets into targeted and untargeted. The targeted tweet refers to a particular person or organization or group of people. Results show that BERT model performs better than other models. Among the SemEval-2020 teams participated for task 12, we were ranked 72 in subtask A and 36 in subtask B.

Model used	F1 Macro
1D-CNN	0.291
2D-CNN	0.3145
BiLSTM	0.2894
BERT	**0.38938**

Table 3: Results for models used in subtask B

5 Conclusion and Future work

There is an increase in the usage of profanity words in online communications due to the ease with which speakers can remain anonymous. People comment about a particular person or an organization or a group of people in an aggressive manner in social media. Due to the fast transmission of online communications, this information is spread rapidly. This has led to the need to detect the offensive tweets and to remove and stop them from spreading further.

SemEval-2020 Task 12 involves three subtasks in which we have participated in subtasks A and B. Deep learning models with different word embeddings were used to perform the tasks. Results show that deep network model with BERT embeddings performs better when compared to GloVe and Word2Vec embedding models. Since the BERT model is based on multilingual case-based representation, this can also be applied to other languages like Tamil, Hindi, Telugu, Kannada, Arabic, Danish, Turkish and Greek. We would like to do further investigation on applying this model to other languages.

Acknowledgements

We would like to thank SSN management for providing us with the infrastructure to carry out this work. This work is carried out in the Machine Learning Research Group (MLRG) GPU server present in the Department of CSE, SSN College of Engineering.

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain, July. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of ICWSM*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

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

Matthew Honnibal and Ines Montani. 2017. Spacy 2: Natural language understanding with bloom embeddings. *convolutional neural networks and incremental parsing*, 7(1).

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.

Ping Liu, Wen Li, and Liang Zou. 2019. Nuli at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91.

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.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 14–17.

Tomas Mikolov, Kai Chen, Gregory S Corrado, and Jeffrey A Dean. 2015. Computing numeric representations of words in a high-dimensional space, May 19. US Patent 9,037,464.

John Pavlopoulos, Nithum Thain, Lucas Dixon, and Ion Androutsopoulos. 2019. Convai at SemEval-2019 task 6: Offensive language identification and categorization with perspective and bert. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 571–576.

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, Doha, Qatar, October. Association for Computational Linguistics.

S Rajalakshmi, Angel Suseelan, B Logesh, S Harshini, B Geetika, S Dyaneswaran, S Milton Rajendram, and TT Mirnalinee. 2019a. Techssn at SemEval-2019 task 6: Identifying and categorizing offensive language in tweets using deep neural networks. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 753–758.

S Rajalakshmi, Angel Suseelan, S Milton Rajendram, and TT Mirnalinee. 2019b. Ssn-sparks at SemEval-2019 task 9: Mining suggestions from online reviews using deep learning techniques on augmented data. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1237–1241.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. In *arxiv*.

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.

Alexandra Schofield and Thomas Davidson. 2017. Identifying hate speech in social media. *XRDS: Crossroads, The ACM Magazine for Students*, 24(2):56–59.

Bhargav Srinivasa-Desikan. 2018. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

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*.

Zhenghao Wu, Hao Zheng, Jianming Wang, Weifeng Su, and Jefferson Fong. 2019. Bnu-hkbu uic nlp team 2 at SemEval-2019 task 6: Detecting offensive language using bert model. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 551–555.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1415–1420.

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*, pages 75–86.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of 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.