2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) | 978-1-7281-7741-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICACITE51222.2021.9404736

Detecting Cyberbullying Across Multiple Social Media Platforms Using Deep Learning

Maheep Mahat
Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India
maheepmahat1@gmail.com

Abstract – In today's time social media platforms have taken over our lives. The number of people using these platforms keeps on increasing day by day. With the increase of social media usage, the person who is using these platforms become more exposed to the negative effects of using social media. Among many negative effects, cyberbullying is one of the major negative effects of using social media. People online get bullied which affects their mental health in a negative manner. It is an incredibly difficult task to detect cyberbullying on social media platforms especially due to the slangs that people make up regularly but even so this paper suggests a working implementation of an application that detects cyberbullying across multiple social media platforms using the data provided by Twitter, Wikipedia and Formspring. This paper makes use of Deep Learning for the purpose of detecting cyberbullying.

Keywords – Deep Learning, Neural Networks, LSTM, Modular Network, Social Media Platforms

I. Introduction

Cyberbullying or cyber harassment is defined as a form of bullying that takes place through the use of electronic communication. It has become increasingly common between people within the age range of 13 to 22. As people's presence increases online, cyberbullying has been seen to increase accordingly. There are various surveys that conclude that from 10% of internet users up to 40% of internet users have been a victim of cyberbullying [1]. The negative effects of cyberbullying have been seen from having anxiety to having suicidal thoughts.

In June of 2020, Indian actress Kangana Ranaut received different kinds of threats and in large number for speaking about negative side of her job.

It is a very difficult task to detect cyberbullying. People in social media platforms don't exactly use perfect English to communicate. A lot of what people write is usually a shortened version of the actual word.

People also use a lot of sarcasm which makes the detection of cyberbullying a very difficult task since a sarcastic comment can easily be taken as a comment intended to demean someone. Across multiple social media platforms, people get attacked based on their race, gender, appearance, etc. Detecting cyberbullying is not the same across different platforms. For example, in Twitter, the word 'bitch' is closely associated with 'female' but there doesn't seem to be similar bias in other datasets. One of the key aspects of detecting cyberbullying across different platform is platform specific semantic similarity. Different platforms have different style of communication. While a platform like Twitter has short paragraphs, which are at maximum of 280 words, other platforms that use the question answer style of communication don't have such limit and also provide option of being anonymous.

Different machine learning approaches were experimented with for detecting cyberbullying. Such as, SVM, CNN, LSTM, naïve Bayes and logistic regression. Among them LSTM had the best results so the final implementation was done using LSTM. While experimenting with all these approaches, following conclusions were drawn.

1) Just having swear words in a piece of text isn't enough of classify the text as intended for bullying.
2) Deep Learning models gave better results compared to traditional machine learning models for the purpose of cyberbullying.
3) Different platforms have different interpretation of words and a piece of text that classifies as cyberbullying in one platform doesn't classify as cyberbullying in another platform.
4) The context of the conversation holds highest weight while analyzing a piece of text. For example, a conversation among friends that has a lot of sarcastic comments about each other, without context, it will classify as bullying.

II. Related Work

Cyber bullying has been recognized as a problem since 2003 [2]. The first time an auto detection approach was implemented was in 2009 [3]. Multiple Machine Learning algorithms have been used for detecting cyber bullying. Among them have been SVM, Logistic regression, Naïve Bayes, Decision Trees as well as approaches that uses Deep Learning to detect cyberbullying [4].

Most of the existing approaches [5] use the following pattern: Collect training data from a single platform, engineer different features with a specific style of cyberbullying as the goal, use some machine learning approach, and take metrics such as F1 scores [6] to evaluate accuracy. These approaches rely heavily on handpicked features such as the use of foul language. These approaches yield lower success rate for cyberbullying detection [7][8] because handpicked features are not completely reliable against different styles of different platforms and bullying subjects.

III. Methodology

A. *Dataset used*

Three datasets are used from three different platforms. Wikipedia, Twitter and Formspring. From each source, 3000 examples were used. With a total of 9000 examples for the training of our model.

B. *Data preprocessing*

The noncontributory characters such as punctuation marks, blank spaces, symbols, numbers and other various text that do not contribute towards classifying a piece of text as bullying were eliminated from the dataset. Mini batch gradient descent is used to train the model. In each batch, sentences with fixed length of 70 were used. In case of sentences with lesser length, the sentences were padded to keep the length constant. Sentences of length greater than 70 were truncated to 70. Each dataset consists of 1500 negative examples and 1500 positive examples.

C. Model Architecture

The same architecture was used for all dataset. Twitter Model was trained on twitter dataset, the Wikipedia model was trained on Wikipedia dataset and the Formspring model was trained on Formspring dataset. A modular network [9][10][11] approach was implemented.

The model architecture [12] comprised of an input layer after which there is a SSWE embedding layer followed by a LSTM [13][14] layer where the return sequences were made to be positive. One more LSTM layer but on this one, the return sequences are made to be negative and a dropout layer having the probability of 0.5 then followed by a dense layer containing two output classes that makes use of the softmax activation function.

Since we need the hidden states within the LSTM at every level, we chain the two LSTMs together. The initial layer of the LSTM gives the hidden state to each time step of the input, after which on different instance, the hidden state output is coupled with the state of the cell for the last time step of the input. All of these steps are coalesced to give a single prediction. Keras backend functions are used to get the hidden state outputs from the initial LSTM layer of each and every model. The LSTM layer returns back three hidden state in array format. The initial one is the LSTM's output of the hidden state for every input time step. The following is the output of hidden state for the preceding time step. The one after that is for the cell state of the final time step. A custom keras layer, provided by each model, receives each hidden state as a coalesced input. For each hidden state, there exists a custom keras layer. From each model, the inputs are combined and a single hidden state is given as output.

By combining outputs of each model, we get three hidden states. The new LSTM layer is given these hidden states as an input. For each step of the time, the merged output of the hidden state is given to a newly constructed LSTM layer. This new layer hidden state is set up with the output of the hidden state that are merged for the final time step and then coalesced cell state. The initial custom layer's result is interlinked with the LSTM layer. This layer is set up using the results from the initial and third custom layer. The LSTM layer connects to dropout layer which has the probability of 0.5. Finally, there is a dense layer that uses softmax activation function. This layer has two classes that are labeled positive and negative. The model uses Adam optimizer and also makes use of the Binary cross entropy loss function.

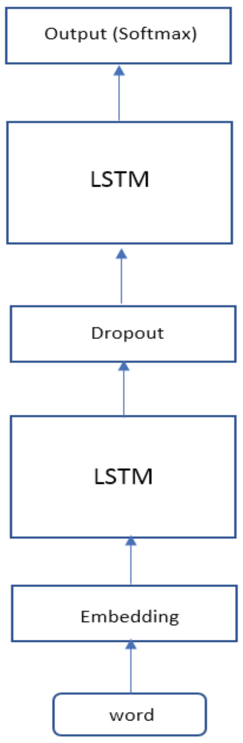

Figure 1 - Model Architecture

IV. Experiments and Result

Here are the results using traditional machine learning algorithms.

Table 1- Outcomes by classic ML algorithms by means of F1 score.

Dataset	label	Character n-grams				Word unigrams			
		LR	SVM	RF	NB	LR	SVM	RF	NB
Formspring	bully	0.448	0.422	0.298	0.359	0.489	0.463	0.264	0.025
Twitter	racism	0.723	0.676	0.752	0.686	0.738	0.772	0.739	0.617
	sexism	0.729	0.688	0.720	0.647	0.762	0.758	0.755	0.635
Wiki	Attack	0.694	0.677	0.674	0.655	0.711	0.686	0.730	0.659

These models were trained making use of the backpropagation approach. Adam optimizer is used in conjunction with the categorical cross-entropy loss function.

Using the above given approaches, there were unsatisfactory results in multiple ways. For example, the

The model was not always able to correctly classify sentences that were a bit tricky. Example – "I hate that you are so better than me at literally everything". With the deep learning approach, the results were much better compared to the traditional approach. The model was properly able to classify tricky sentences as mentioned above. For the testing purpose, 1000 test examples were taken. The test dataset was used on each of the individual model. Here are the results for the different datasets.

Table 2 – Accuracy results on different datasets

Wikipedia	Twitter	Formspring
75.5%	79.1%	72%

While testing our model, the test data goes through each and every model to build hidden states which then proceed onto the transitional parts for the final result. The accuracy of 77.9% is achieved using this model.

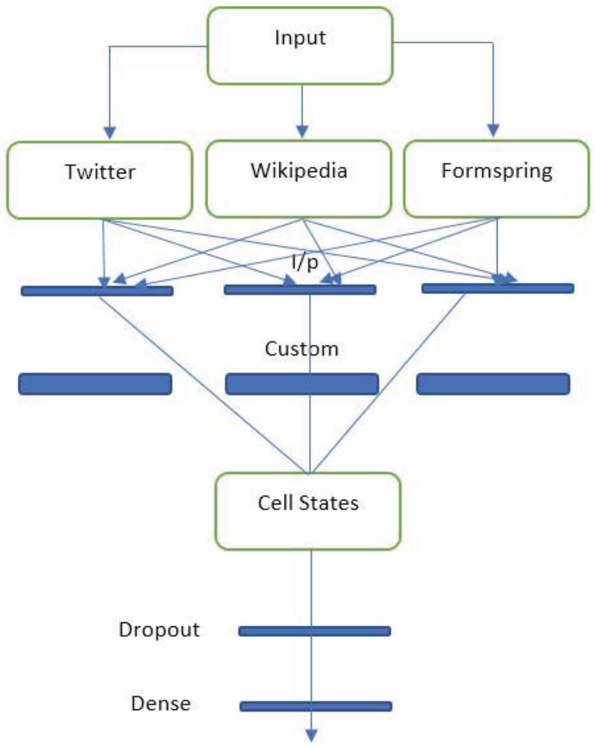

Figure 2 – Proposed Modular architecture

Another approach we could take is by merging just the hidden state output from the final time step and cell state for the final time step from every model. Then we could initialize hidden states of some other LSTM and those states. After that, we could train the LSTM on the datasets successively.

V. Conclusion

To conclude, this approach was able to get better results compared to the traditional approach for detecting cyberbullying [15][16]. It was also able to outperform deep learning approaches where data from a single source was used to train a single model and approaches that rely entirely on word embeddings to understand text data [17]. Those models did not perform good when they were fed data from other sources. The results can be improved further by improving the dataset by having as much relevant data as possible and reducing the redundant data. The hyper parameters of the model could also be improved for increasing the accuracy of the model. Working on these mentioned areas of improvement is kept for future work. By improving on these things, we can achieve even better results.

VI. References

[1] Whittaker and R. M. Kowalski. Cyberbullying via social media. Journal of School Violence, 14(1):11–29, 2015

[2] R. L. Servance. Cyberbullying, cyber-harassment, and the conflict between schools and the first amendment. Wisconsin Law Review, pages 12–13, 2003.

[3] Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards. Detection of harassment on web 2.0. In The workshop on Content Analysis in the WEB 2.0, WWW, pages 1–7, 2009.

[4] Rekha Sugandhi, Anurag Pande, Siddhant Chawla, Abhishek Agrawal, Husen Bhagat, “Methods for Detection of Cyberbullying : A Survey” in IEEE December 2015.

[5] R. Johnson and T. Zhang. Supervised and semi-supervised text categorization using LSTM for region embeddings. In ICML, pages 526–534, 2016

[6] Goutte, Cyril and Éric Gaussier. “A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation.” *ECIR* (2005).

[7] S. Hinduja and J. W. Patchin. Bullying, cyberbullying, and suicide. Archives of suicide research, 14(3):206–221, 2010.

[8] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In ACL, pages 1555–1565, 2014

[9] Gasser Auda, Mohamed Kamel, "MODULAR NEURAL NETWORKS: A SURVEY", International Journal of Neural SystemsVol. 09, No. 02, pp. 129-151 (1999).

[10] Bart L.M. Happel, Jacob M.J. Murre, "The Design and Evolution of Modular Neural Network Architectures", Neural Networks, 1994, 7, 985-1004.

[11] Modular neural network – https://en. wikipedia. org/ wiki/ Modular_neural_network

[12] Bart L.M. Happel, Jacob M.J. Murre, "The Design and Evolution of Modular Neural Network Architectures", Neural Networks, 1994, 7, 985-1004

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. Neural Comput. 9, 8 (November 15, 1997), 1735–1780. DOI:https://doi.org/10.1162/neco.1997.9.8.1735

[14] Sepp Hochreiter, Jurgen Schmidhuber, "LONG SHORT-TERM MEMORY" in Neural Computation 9(8):1735{1780, 1997

[15] Whittaker and R. M. Kowalski. Cyberbullying via social media. Journal of School Violence, 14(1):11–29, 2015.

[16] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. In COLING, pages 3485–3495, 2016.

[17] Introduction to Word Embedding and Word2Vec - https://towardsdatascience. com/ introductionto-word-embedding-and-word2vec-652d0c2060fa