IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Special Issue 2, March 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

| International Conference on Recent Innovations in Engineering and Technology (ICRIET'23)|

| Sharadchandra Pawar College of Engineering, Pune, India |

|| Volume 11, Special Issue 2, March 2023 ||

Using Small BERT Models, a Quick Method for Identifying Cyberbullying

Rushikesh Ambre , Yogendra Yadav, Nikhil Kamble , Gaytri Khule, Dr. Sunil Khatal

Students, Department of Computer Engineering, Sharadchandra Pawar College of Engineering, Pune, Maharashtra, India

HOD, Department of Computer Engineering, Sharadchandra Pawar College of Engineering, Pune, Maharashtra, India

ABSTRACT: Many academics have been focusing on the problem of identifying cyberbullying because many people now use social media to spread hatred. the previous ten years. Transfer learning is employed in this study to resolve this problem. We use information from hatred speech to modify our small BERT models. To correct the data's class disparity, we employ the Focus Loss function. On the hate speech dataset, we were able to achieve ground-breaking results using this technique, including 0.91 accuracy, 0.92 recall, and 0.91 F1-score. We also show that the more compact BERT models are much quicker at identifying cyberbullying and are suitable for real-time applications using our transfer learning process.

KEYWORDS: Cyberbullying, Focus Loss, Learning Techniques, Hate Speech, Concise BERT

1. INTRODUCTION

Numerous real-world issues have been brought to light by the social media industry's rapid growth over the previous ten years on the internet. Hatred and bullying have long been problems in civilization. lengthy time. But these abusers can now discretely send irate, derogatory, or offensive messages to a single person or group of people while hiding behind a laptop, cell phone, or social networking site. Cyberbullying affects people, and many children and adolescents who engage in it struggle with melancholy. [1] If Cyberbullying was identified as soon as it showed online, it would be very beneficial. As a result, reports of harassment have increased in frequency over the past few months on various social media platforms.

As one tries to resolve this problem, they encounter many difficulties and challenges. The use of informal language, the use of emojis, the use of various languages, the absence of a robust benchmark data set, and the necessity for quick real-time recognition in the streaming data are a few major issues [2]. In this article, we place a focus on hastening the discovery of cyberbullying. study and demonstrate how transfer learning strategies allow for the functional parity of smaller networks with bigger ones. We contribute twice to this area of research. First, we tweak a few small BERT models [3] to increase the detection speed of cyberbullying and achieve cutting-edge performance. Second, by using the Focal Loss function to tweak BERT models, we show how this function can be used to further enhance these models.

II.RELATED WORK

Many academics have been working on a method to identify harassment for the past ten years. The textual features from early attempts like [4] and [5] were used to train type classifiers like SVM or Naive Bayes by extracting textual features using more conventional natural language processing methods like N-grams and TF-IDF. In actuality, many excellently written and interesting articles are listed in surveys like [2]. Recurrent and convolutional neural networks (RNN and CNN), among others, have significantly contributed to the development of deep learning methods. Simulated language. In order to handle the problem of identifying cyberbullying, many approaches, including [6][7][8], created different LSTM and CNN models.

The word embedding layers used in these methods, such as word2vec[9] or Glove[10], are usually pre-trained on a large collection of words. These layers map each word into a top vector where words with related meanings are clustered close to one another. Some techniques, like [8], incorporate user information into their identifying process, such as the user's buddy network and number of followers. A hybrid classifier with a text route and a meta data path is learned by the experts. Over the past few years, this topic has been the focus of a lot of tasks and competitions.

In reality, teams that participated in events like SemEval2019 produced a number of written works [11]. You might notice a pattern of articles like [12] and [13] using Transformer-based designs like BERT [14]. In reality, 7 of the top 10 teams in the SemEval2019 challenge for offensive language detection used BERT-based designs. [11] BERT's transformer levels allow for a significant degree of parallelization. [15] The advantage of parallelization is performance improvement. BERT pre-trained models are also effective language representation models that are easy to refine and produce cutting-edge results. [14]

III.SUGESTED METHOD

A. *Distribution of Documents*

THE DATA COLLECTION ON HATE STATEMENTS COLLECTED IN [16] WAS USED IN THIS INVESTIGATION. 85948 TWEETS THAT HAVE BEEN PUBLICLY TAGGED ARE INCLUDED IN THIS DATA COMPILATION. THE AVERAGE, AGGRESSIVE, AND HATEFUL TARGET GROUPS ARE DEFINED. FIGURE 1 ILLUSTRATES THAT MOST OF THE DATA IS NEITHER RUDE NOR OFFENSIVE. FURTHERMORE, THE DATA COLLECTION IS SMALL AND THE DISLIKING CLASS IS SMALL.

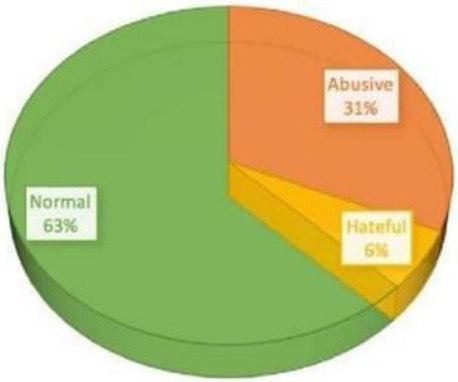

Fig. 1. Hate-Speech Data Distribution

B. *Preparation of Content:*

Frequently, symbols and hashtags are used in Twitter content to link to other sites. The @usernames were converted to the word "URL" and vice versa, replacing the links initially. using reduce Then, inspired by [12], we decided to use the helpful information masked by emoticons and hashtags. Since hashtags can include a large number of words or even a complete phrase, we used a Python tool that is available as open source on GitHub. Using a programme called Word segment, the codes were broken up into sentences.

1. The word "#drawntodeath" will become "drawn to death" as a result of this segmentation procedure. We made use of a separate open-source, free Python tool from There is a secret emoji there because GitHub transformed every occurrence of an emoji using Emoji

2. For instance, a furious emoticon Finally, we took care to lowercase each capital character. The better BERT models require this change and only use uncased text for training.

C. *Tiny BERT Figures*

Bidirectional Encoder Representations from Transformer, also known as BERT, can be used to address a wide range of natural language issues, including sentiment analysis and text categorization with fine-tuning [14]. There is a disadvantage to the initial BERT, and it has to do with dimensions. Or to put it another way, BERT is a big network with lots of transformer levels and hidden embeddings. Therefore, with lesser data sets, fine-tuning would not result in the finest results.

Recently, compact BERT models that solve this problem were made available [3]. The researchers produced a total of 24 small BERT models with various concealed embedding sizes and transformer layer counts. Each of these networks was taught using an instructor network, which was essentially a very large BERT pre-trained model. To give the pupil network access to the teacher's soft labels, they used the distillation method and unlabelled data.

We selected 5 of these 24 tiny BERT models for the tests in this research. With transformer layer numbers varying from 2 to 12, and hidden embedding sizes ranging from 128 to 768, Table I demonstrates the variety of the designs selected.

TABLE I
COMPACT BERT ARCHITECTURES THAT WERE INTRODUCED IN [3]

Model Name	Transformer Layers	Hidden Embedding Sizes
BERT-Base	12	768
BERT-Medium	8	512
BERT-Small	4	512
BERT-Mini	4	256
BERT-Tiny	2	128

D. *Monitoring pipeline*

The workflow we use to divide the processed data into the three groups of "normal," "abusive," and "hateful" is shown in Figure 2. The prepossessed data is loaded in its totality at first as groups of text and actual labels. Text occurrences are padded if required to make the series duration. The content is then tokenized using a pre-trained BERT tokenizer.

Each pre-trained BERT model that later produces a token lexicon comes with a pre-defined vocabulary collection. The pre-trained BERT tokenizer converts text from a series of words into a sequence of numeric Identifiers using this token vocabulary.

The concluding layer of the BERT model is removed, and in its stead We include a dense layer with a dimension of three because there are three distinct groups. a Plush The cohort with the greatest likelihood number will choose the ultimate predicted label.

E. *Focal Loss*

We chose to use Focal Loss as our cost function because the work of [17] encouraged us to do so. Initially outlined in [18], Focal Loss is a type of Cross-Entropy loss that also takes into account how easy or difficult it is to identify each sample.

Programs that struggle with class inequality have shown that it is useful [18]. Equation 1 shows how to calculate focus loss, with pt = p indicating that the sample belongs to the positive class and that y = 1 is the proper designation. If not, pt equals 1 less p. This idea states that a sample has a lower pt if it is easier to categorise.

$$F\ L(pt) = -\alpha t(1 - pt)\ \gamma \log(pt) \ldots\ldots\ldots\ldots (1)$$

It is essential to choose the right hyper-parameter for each application. In the part that follows, we'll go over how this number was selected for use in our case.

IV.TESTS AND RESULTS

F. **SETUP FOR THE TRAINING:**

To develop this project, we used Kera's in the Google Collaboratory setting. We could use the TPU motors. We chose to use AdaBound [19], a comparatively recent optimization strategy that can lead to more fluid training. Table II lists each hyper-parameter used in our configuration.

TABLE II
TRAINING HYPER-PARAMETERS

Batch Size	128
Sequence Length	128
Number of Epochs	5
Learning Rate	0.0001
Focal Loss Parameter γ	0.1

G. FL Hyper-parameter Decision

In order to obtain the optimal number for our Focal Loss application, we split the data at random into 90% train and 10% validation. The best test results were then determined by adjusting to several values using Small-Bert and fixing every other hyper-parameter. It's important to keep in mind that using the default Cross-Entropy loss corresponds to putting = 0.

As shown in Table III, using values for that are excessively big resulted in lower assessment measures. This happens because as the number of samples rises, the weight of easy-to-classify samples lowers, which can be detrimental to the training process. The best result appears to be for = 0.1, which is significant but not so significant as to cause cases that are easy to categories to be ignored.

TABLE III
IMPACT OF γ ON VALIDATION RESULTS

γ	Accuracy	AUC	Precision	Recall	F1-score
0	0.9092	0.9702	0.9003	0.9092	0.9021
0.01	0.9138	0.9705	0.9062	0.9138	0.9077
0.1	0.9143	0.9709	0.9064	0.9143	0.9076
1	0.9125	0.9700	0.9029	0.9125	0.9033
2	0.9145	0.9683	0.9063	0.9145	0.9064
5	0.9113	0.9654	0.9026	0.9113	0.9026
10	0.9077	0.9643	0.8991	0.9076	0.8955

H. Analysis's Results

We used 10-fold cross validation on the complete data set to assess performance and properly compare our final results to previous work by Fountas et al. [8]. We used a variety of assessment factors to identify the best model, paying particular attention to the F1-score, which stands for the harmonic mean Precision and Recall.

Table IV demonstrates that our approach can deliver outcomes that are better than those of previous studies on the same data collection. Even though our approach ignores the user-based and network-based information that Fountas et al.[8] employ, the increase is still achieved.

It's also interesting to see how well the assessment metrics for these tiny BERT models line up. BERT-Base is the best model for our task if we only consider the measures, as it has the greatest F1-score.

TABLE IV
EVALUATION RESULTS

Model	Accuracy	AUC	Precision	Recall	F1-score
BERT-Base	0.9156	0.9734	0.9090	0.9156	0.9103
BERT-Medium	0.9140	0.9726	0.9071	0.9140	0.9084
BERT-Small	0.9147	0.9722	0.9080	0.9147	0.9093
BERT-Mini	0.9148	0.9717	0.9078	0.9148	0.9086
BERT-Tiny	0.9147	0.9699	0.9066	0.9147	0.9064
Founta et al. [8]	0.84	0.93	0.85	0.85	0.85

But we also considered the time required to teach and evaluate each network. On Google Collaboratory TPU with 8 employees, time was determined based on how long it took to handle a batch of data, which was set to 128 for both the train and test phases.

Table V, the results of our time study, demonstrates that the models' velocities differ quite a bit, with training periods exhibiting more variance than test times. Although it was once believed that adding more transformer layers and concealed embedding sizes would slow down the networks, the results of the assessments were actually noticeably improved.

This is also untrue, as BERT-Tiny, which travels at the speediest rate of 6 Ms per step, trails BERT Base, which moves more slowly at 17 Ms per step, in the F1 ranking by just 0.04 p.c. It is fair to say that in this case, more compact networks would have more to offer if they were used in a system that needed real-time monitoring.

TABLE V
COMPACT BERT MODELS TIME ANALYSIS

Model	Training Time	Test Time
BERT-Base	136ms	17ms
BERT-Medium	65ms	10ms
BERT-Small	40ms	7ms
BERT-Mini	29ms	7ms
BERT-Tiny	19ms	6ms

V.CONCLUSION AND FUTURE WORK

Using transfer learning and enhanced compact BERT models, we presented a new method for identifying cyberbullying in this research. We were able to beat previous work without using any information. Additionally, we demonstrated that our technology is efficient and reliable, making it perfect for real-time harassment detection.

REFERENCES

1. Rushikesh Ambre1, Yogendra Yadav2, Nikhil Kamble3, Gaytri Khule4, Dr. Sunil Khatal5,'Rapid technique for detecting cyberbullying using Small BERT Models', International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume 2, Issue 1, November 2022.
2. M. P. Hamm, A. S. Newton, A. Chisholm, J. Shulhan, A. Milne, P. Sundar, H. Ennis, S. D. Scott, and L. Hartling, "Prevalence and effect of cyberbullying on children and young people: A scoping review of social media studies," JAMA pediatrics, vol. 169, no. 8, pp. 770–777, 2015.
3. S. Salawu, Y. He, and J. Lumsden, "Approaches to automated detection of cyberbullying: A survey," IEEE Transactions on Affective Computing, 2017.
4. I. Turc, M.-W. Chang, K. Lee, and K. Toutanova, "Well-read students learn better: On the importance of pretraining compact models," arXiv preprint arXiv:1908.08962v2, 2019.
5. M. Dadvar and F. De Jong, "Cyberbullying detection: a step toward a safer internet yard," in Proceedings of the 21st International Conference on World Wide Web, 2012, pp. 121–126
6. A. Kontostathis, K. Reynolds, A. Garron, and L. Edwards, "Detecting cyberbullying: query terms and techniques," in Proceedings of the 5th annual acm web science conference, 2013, pp. 195–204. [6] P. Singh and S. Chand, "Pardeep at semeval-2019 task 6: Identifying and categorizing offensive language in social media using deep learning," in Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 727–734.
7. V. Golem, M. Karan, and J. Snajder, "Combining shallow and deep ˇ learning for aggressive text detection," in Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 188– 198.
8. A. M. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, and I. Leontiadis, "A unified deep learning architecture for abuse detection," in Proceedings of the 10th ACM Conference on Web Science, 2019, pp. 105–114.
9. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

10. J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
11. M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval)," arXiv preprint arXiv:1903.08983, 2019.
12. P. Liu, W. Li, and L. Zou, "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, 2019, pp. 87–91.
13. P. Aggarwal, T. Horsmann, M. Wojatzki, and T. Zesch, "Ltl-ude at semeval-2019 task 6: Bert and two-vote classification for categorizing offensiveness," in Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 678–682.
14. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.
15. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in neural information processing systems, 2017, pp. 5998–6008.
16. A.-M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis, "Large scale crowdsourcing and characterization of twitter abusive behavior," in 11th International Conference on Web and Social Media, ICWSM 2018. AAAI Press, 2018.
17. S. Srivastava, P. Khurana, and V. Tewari, "Identifying aggression and toxicity in comments using capsule network," in Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 98–105.
18. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss ´ for dense object detection," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
19. L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," arXiv preprint arXiv:1902.09843, 2019.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.379

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details