IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

Expeditious Cyber-Bullying Detection Method Utilizing Compact BERT Models

Rushikesh Ambre[1], Yogendra Yadav[2], Nikhil Kamble[3], Gaytri Khule[4], Dr. Sunil Khatal[5]
Students, Department of Computer Engineering[1,2,3,4]
Professor, Department of Computer Engineering[5]
Sharadchandra Pawar College of Engineering, Pune, Maharashtra, India

Abstract: *Since many people now use social media to disseminate hate, many researchers have been concentrating on the issue of detecting cyberbullying. in the last ten years. In this work, transfer learning is used to address this issue. We adjust our tiny BERT models using data from hate speech. We use the Focal Loss function to address the data's class imbalance. With this method, we were able to obtain cutting-edge outcomes on the hate speech dataset, including 0.91 precision, 0.92 recall, and 0.91 F1-score. Additionally, we demonstrate using our transfer learning pipeline that the more compact BERT models are much faster at detecting cyberbullying and are appropriate for real-time applications.*

Keywords: Focal Loss, Transfer Learning, Hate Speech, Compact BERT, Cyber bullying

I. INTRODUCTION

Due to the social media industry's explosive expansion over the past ten years, many real-world problems have the world wide web. Bullying and hatred have been issues in society for a while. long period. But such bullies can now effortlessly slink away behind a using a computer, smartphone, or social media platform, send angry, insulting, or offensive texts to another person or a collection of people. This practise known as cyberbullying, has an impact on individuals and many youngsters experience depression and teenagers.[1] It would be extremely advantageous if Cyberbullying was discovered as soon as it appeared online. Due to this, attention has been drawn more frequently to cyberbullying. detection during the past few months on various social media years.

One faces numerous obstacles and challenges as they attempt to solve this issue. A few significant problems are the use of casual language, the use of emojis, the use of different languages, the lack of a strong benchmark data set, and the requirement for rapid real-time identification in the streaming data [2]. We emphasise speeding up the detection of cyberbullying in this work and show that transfer learning approaches enable smaller networks to function on par with larger ones.

We make two contributions to this field of study. First, in order to improve the detection speed of cyberbullying and attain cutting-edge performance, we fine-tune a number of tiny BERT models [3]. Second, we demonstrate how the Focal Loss function can be used to improve these models further by using it to fine-tune BERT models.

II. RELATED WORK

Over the past ten years, a lot of researchers have worked to find a solution to the detection of cyberbullying. Early efforts like

[4] and [5] used more traditional natural language processing techniques like N-grams and TF-IDF to extract textual features, which were then used to train type classifiers like SVM or Naive Bayes. In truth, there are a lot of well-written, fascinating pieces that are included in polls like [2].

Neural networks, such as recurrent and convolutional neural networks (RNN and CNN), have played a significant part in deep learning methodologies' later impetus. linguistic simulation. As a result, numerous approaches, including [6][7][8], developed various LSTM and CNN models to address the issue of detecting cyberbullying. These techniques also use word embedding layers, typically pre-trained on a huge set of words, like word2vec[9] or Glove[10]. In a place where words with similar meanings are located near to one another, these layers map each word into a high-dimensional vector. Some methods, like [8], also use user metadata into their identification process, such as the user's network of friends and the amount of followers they have. A combination classifier with a text path and a metadata path is trained

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

by the researchers. This subject has been the subject of numerous contests and challenges over the last couple of years. In fact, a number of works that have been published come from teams that took part in competitions like SemEval2019[11]. You may see a trend toward employing Transformer-based architectures like BERT[14] in several papers like [12] and [13]. In fact, BERT-based architectures were used by 7 of the top 10 teams in SemEval2019's offensive language detection competition.

Transformer layers in BERT enable a large amount of parallelization. [15] Parallelization increases speed, which is a benefit. Furthermore, BERT pre-trained models are potent language representation models that are simple to hone and yield cutting-edge outcomes. [14]

III. SUGESTED METHOD

3.1 Data Distribution

This experiment made use of the hate-speech data set compiled in [16]. This data collection consists of 85948 tweets that have been labelled by the public. Three target classes—Normal, Abusive, and Hateful—are identified. The majority of the data is neither abusive nor nasty, as can be shown in Figure 1. Additionally, the hating class is tiny and the data set is

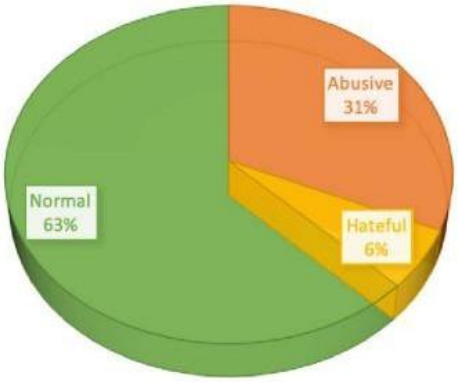

Fig. 1. Hate-Speech Data Distribution

3.2 Text Preparation

Typically, Twitter's content also includes emoticons and hashtags. as a connection to other pages. The links were first swapped out for the Every @username was changed to the term "URL" and vice versa. ”use rid”. After that, motivated by [12], we chose to use the useful data concealed in emojis and hashtags. Since hashtags can consist of many words or even an entire sentence, GitHub has an open-source Python library that we used. The hashtags were divided into words using a tool called Word segment

1. For instance, the hashtag "#drawntodeath" will after this segmentation process, get "drawn to death." We utilised a different free and open-source Python library from Each instance of an emoji was converted by GitHub using Emoji2 to there is a hidden emoji there. For instance, an emoji depicting angry Last but not least, we made sure to lowercase every uppercase letter. Only employing uncased text for training, the BERT models we improved necessitate this modification.

3.3 Small-size BERT Models

Sentiment analysis and text classification with fine-tuning are only two examples of the many natural language problems that can be solved with Bidirectional Encoder Representations from Transformer, also known as BERT [14]. The original BERT does have a drawback, and that drawback is related to size. In other words, BERT is a sizable network with numerous transformer layers and covert embeddings. Therefore, fine-tuning would not produce the best outcomes with smaller data sets. Compact BERT models that address this issue were just recently released in [3]. A total of 24 compact BERT models were created by the researchers, with different hidden embedding sizes and the

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

number of transformer layers. A teacher network that was effectively a very big BERT pre-trained model was used to train each of these networks. They employed the distillation approach and unlabelled data to enable the student network to benefit from the teacher's soft labels.

Of these 24 tiny BERT models, we chose 5 for this study's experiments. Table I shows how diverse the chosen topologies are, with transformer layer counts ranging from 2 to 12 and hidden embedding sizes from 128 to 768.

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

3.4 Pipeline for Detection

In Figure 2, you can see the pipeline we utilise to classify the processed data into three categories: normal, abusive, and hateful. The entirety of the prepossessed data is initially loaded as batches of text and real labels. If necessary, text instances are padded to meet the sequence length. A pre-trained BERT tokenizer is then used to tokenize the text.

A pre-defined vocabulary set is provided with each pre-trained BERT model that subsequently generates a token dictionary. This token dictionary is used by the pre-trained BERT tokenizer to transform text from a sequence of words into a sequence of numeric IDs.

The BERT model's final layer is dropped, and in its place Because there are three different classes, we include a dense layer with a size of 3. To provide probability scores for each class, a SoftMax layer is applied after the dense layer. The final predicted label will be determined by the class with the highest likelihood score.

3.5 Focal Loss

We made the choice to adopt Focal Loss as our cost function after being inspired by the work of [17]. Focal Loss, a kind of Cross-Entropy loss that also considers how simple or difficult it is to classify each sample, was initially described in [18]. Applications that have an issue with class imbalance have demonstrated it to be advantageous [18]. Equation 1 illustrates the computation of focal loss, where pt = p if the sample belongs to the positive class and the correct label is y = 1. If not, pt = 1 minus p. According to this concept, a sample has a smaller pt if it is simpler to classify.

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

For each application, selecting the appropriate hyper-parameter is crucial. We will describe how this parameter was chosen for our situation in the following section.

IV. TESTS AND OUTCOMES

4.1 Setup for the Training

We used Kera's in the Google Collaboratory environment to create this project. The TPU engines could be used by us. We decided to employ AdaBound [19], a relatively new optimization approach that can result in smoother training. Table II shows all of our setup's hyper-parameters.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

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

4.2 FL Hyper-parameter Decision

We randomly divided the data into 90% train and 10% validation in order to get the ideal value of for our application of Focal Loss. Then, we adjusted to several values using Small-Bert and fixed every other hyper-parameter to determine which one would produce the best validation results. It is crucial to remember that utilising the standard Cross-Entropy loss is equivalent to setting = 0.

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

Using numbers for that are excessively large led to lower evaluation metrics, as seen in Table III. This occurs because as increases, the weight of samples that are straightforward to classify decreases, which can harm the training process. The optimum outcome seems to be for = 0.1, which is large enough to have an effect but small enough to prevent disregarding examples that are simple to categorise.

4.3 Evaluation Findings

To evaluate the performance and fairly compare our final findings to earlier work by Fountas et al. [8], we applied 10-fold cross validation to the entire data set. To determine the optimum model, we employed a variety of evaluation criteria, placing more attention on the F1-score, which represents the harmonic mean Precision and Recall.

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

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

As shown in Table IV, our strategy is able to produce results that are superior to those of earlier research on the same data set. The improvement is made even though our method disregards the user-based and network-based metadata that Fountas et al.[8] use.

It's also intriguing to note how well these tiny BERT models match up in terms of evaluation measures. If we merely look at the metrics, BERT-Base is the best model for our work because it has the highest F1-score.

But we also took into account how long it took to train and test each network. Time was calculated based on how long it took to process a batch of data, which was set to 128 for both the train and test stages on Google Collaboratory TPU with 8 workers. Our time study in Table V shows that the models' speeds vary rather considerably, with training times showing more variation than test times. Although it was once thought that increasing the number of transformer layers and hidden embedding sizes would make the networks slower, this actually led to significantly better assessment outcomes. This is not the case, either, as BERT-Tiny, which moves the fastest at 6 milliseconds per step, behind BERT Base, which moves more slowly at 17 milliseconds per step, in the F1 score by just 0.04 percent. Therefore, it is safe to argue that in this situation, if more compact networks were to be used in a system that required real-time detection, they would have more to give.

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

In this study, we introduced a novel technique for detecting cyberbullying that was based on transfer learning and optimised compact BERT models. Without using any metadata, we were able to outperform earlier work. Furthermore, we showed that our technology is quick and accurate, which makes it ideal for real-time cyberbullying identification

ACKNOWLEDGMENT

Based on work made possible by the National Science Foundation via Grant No. 1813858, this information. The Herman P. & Sophia Taubman Foundation made a generous donation to help fund this research as well.

REFERENCES

[1].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.

[2].S. Salawu, Y. He, and J. Lumsden, "Approaches to automated detection of cyberbullying: A survey," IEEE Transactions on Affective Computing, 2017.

[3].I. Turc, M.-W. Chang, K. Lee, and K. Toutanova, "Well-read students learn better: On the importance of pre-training compact models," arXiv preprint arXiv:1908.08962v2, 2019.

[4].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

[5].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.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, November 2022
Impact Factor: 6.252

[6]. 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.

[7]. 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.

[8]. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[9]. 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.

[10]. 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.

[11]. 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.

[12]. 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.

[13]. 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.

[14]. 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.

[15]. 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.

[16].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.

[17]. 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.

[18]. L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," arXiv preprint arXiv:1902.09843, 2019.