Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/52/20241380

Sentiment analysis of Twitter user text based on the BERT model

Chenyang Zhou

School of Software, East China University of Technology, Nanchang, 330000, China

2020213474@ecut.edu.cn

Abstract. Deep Neural Networks (DNNs) utilizing Recurrent Neural Network (RNN) architectures have found extensive application in text sentiment analysis. A prevailing notion suggests that augmenting the model's capacity can significantly improve accuracy and overall model performance. Building upon this premise, this paper advocates the adoption of a larger BERT model for text sentiment analysis. Bidirectional Encoder Representations from Transformers (BERT) is a sophisticated pre-trained language comprehension model that leverages Transformers as feature extractors. However, as the amount of model data increases, exceeding the memory limitations of a single GPU, algorithm optimization becomes crucial. Therefore, this paper employs two methods, namely data parallelism and GPipe parallelism, to accelerate and optimize the BERT model. Compared to a single GPU, training speed almost linearly increases with the addition of more GPUs. In addition, this research investigates the accuracy of the most advanced language model, chatgpt, by reannotating the dataset. During training, it was observed that the accuracy of the chatgpt-annotated dataset significantly declined in both RNN and BERT models. This indicates that chatgpt still exhibits some errors in sentiment text analysis.

Keywords: BERT, Sentiment analysis, Data optimization.

1. Introduction

In the rapidly evolving society, people are immersed in an unprecedented era of media information proliferation. Major platforms like Twitter, Facebook, and Amazon have amassed vast user communities, creating a dynamic digital landscape where individuals actively express their emotions, thoughts, and opinions through comments and interactions. This has given rise to a significant field of research in deep learning: sentiment analysis of user comments. Sentiment analysis acts as a crucial link between the wealth of unstructured user-generated data and valuable insights with broad applications. By discerning sentiment and emotional cues in comments, it reveals patterns and trends with far-reaching implications. These insights are invaluable for businesses in shaping marketing strategies, policymakers in understanding public opinion, and researchers in deciphering human behavior in the digital age.

Contemporary research in text sentiment analysis primarily focuses on preprocessing methods applied to Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) models [1-3]. While these methods excel at keyword extraction, they come with inherent limitations. CNN models, renowned for their keyword extraction capabilities, face difficulties handling lengthy textual content. Their localized perception limits their proficiency with extended sentences. Small convolutional kernels

DOI: 10.54254/2755-2721/52/20241380

in the convolutional layers hinder the capture of extended contextual information, leading to the oversight of crucial dependencies in lengthy sentences and impacting model performance. Similarly, conventional RNN models grapple with vanishing or exploding gradients when dealing with prolonged dependencies. Gradient attenuation or amplification across multiple time steps makes capturing long-range dependencies challenging. This constraint restricts the effectiveness of RNN models in tasks involving lengthy text, extensive sequences, or protracted temporal relationships. Addressing these challenges is crucial to advancing text sentiment analysis methodologies in accordance with the exacting standards of the scientific community.

In order to address the inherent constraints associated with the previously mentioned CNN and RNN models, this paper adopts a preprocessing approach grounded in the principles of the Bidirectional Encoder Representations from Transformers (BERT) model [4]. BERT stands out by capitalizing on the transformative potential of the Transformer architecture and employs a dual-stage process involving pretraining and fine-tuning [5]. The incorporation of BERT preprocessing has been instrumental in achieving noteworthy advancements across various natural language processing tasks [6, 7]. Its unique characteristics have paved the way for substantial performance enhancements, offering a promising avenue for overcoming the limitations that beset traditional CNN and RNN models, as expounded in the preceding discussion.

In this study, a strategic approach was employed to expedite the processing of the BERT model through the utilization of data parallelism and the GPipe technique. The BERT model was partitioned into distinct segments, each assigned to run on separate Graphics Processing Units (GPUs). This partitioning effectively alleviated the memory burden placed on individual GPUs, thereby facilitating the training of more expansive models or accommodating larger batch sizes. It is worth emphasizing that the successful implementation of data parallelism and GPipe techniques necessitat es meticulous engineering efforts and fine-tuning. This includes considerations such as data partitioning, parameter synchronization, and gradient accumulation, among other intricacies. Moreover, the efficacy of the speedup achieved through these techniques is contingent upon a constellation of factors, encompassing the magnitude of the training dataset, the architectural complexity of the model, and the available computational resources. Consequently, it is imperative to conduct empirical experiments and make necessary adjustments, tailored to the specific task and resource constraints at hand, in order to attain an optimal level of training performance [8, 9].

In addition, noting the recent popularity of large language models such as ChatGPT, this paper also conducts research on the accuracy of their emotional processing. After re-annotating the dataset using the large model, its accuracy is tested. It is found that the accuracy has declined, so this study analyzes the reasons and discover that the large model is unable to accurately recognize emotions in some comments.

2. Method

2.1. Data preparation

The research methodology employed in this experiment is depicted in Figure 1. The dataset utilized in this study was sourced from Stanford University and is referred to as Sentiment140 [10]. It consists of 1.6 million tweets collected through the Twitter API, which can be utilized for emotion detection purposes. The dataset used in this study consists of six distinct fields. Firstly, the "target" field indicates the sentiment polarity of the tweet, with a value of 0 representing a negative sentiment and a value of 4 representing a positive sentiment. Secondly, the "ids" field in the dataset serves as a distinctive identifier assigned to each tweet. The "date" field specifies the date when the tweet was posted. The "flag" field denotes the query term associated with the tweet, with a value of "NO_QUERY" indicating that there is no specific query. The "user" field contains the username of the tweet author. Lastly, the "text" field contains the actual content of the tweet. These six fields provide valuable information for conducting sentiment analysis and emotion detection tasks using this dataset.

DOI: 10.54254/2755-2721/52/20241380

Subsequently, the data was cleaned, which involved removing duplicate tweets, handling missing values (if any), and deleting unnecessary information such as ids, date, flag, and user. The sentiment labels were then mapped to binary values, for instance, mapping 0 to negative sentiment and 1 to positive sentiment. This mapping was done to make the dataset suitable for binary classification tasks.

The dataset was partitioned into two sets, with the majority (80%) being allocated for training and the remaining portion (20%) set aside for testing. This separation allowed the model to be trained on a significant portion of the data, while also providing a separate subset for evaluating its performance. The remaining portion of the dataset was then reserved for evaluating the performance of the trained model on unseen data, providing an unbiased assessment of its effectiveness. This partitioning strategy helps in assessing the generalization capability of the model and ensures that it can perform well on new, unseen instances beyond the training data. This division allowed for effective evaluation and validation of the model's performance. Following that, text preprocessing was conducted, which included word segmentation, removing punctuation marks, and converting all text to lowercase, among other steps. Since the BERT model requires input in a specific format for encoding, SentencePiece was employed for word segmentation. The pre-trained word segmenter of BERT was then utilized to convert the text into an input format acceptable to the model.

Specifically, the text was segmented into individual words or subwords using the WordPiece word segmenter. Frequently occurring strings were used as basic vocabulary units, while uncommon words were split into more common parts. Special markers were added to the segmented text, with [CLS] denoting the start of the classification task and [SEP] indicating the end of the sequence. Additionally, the BERT model requires all input sequences to have the same length. If the length of the text is shorter than the required length, the padding marker [PAD] is appended at the end to increase its length. Conversely, if the length of the text exceeds the required length, it can be truncated to meet the required length.

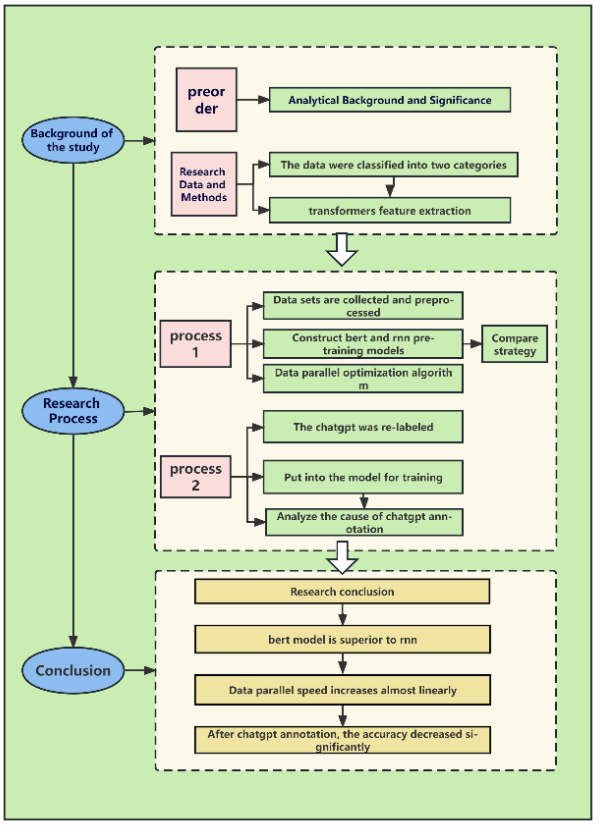

Figure 1. The procedure of this study (Photo/Picture credit: Original).

DOI: 10.54254/2755-2721/52/20241380

2.2. BERT

The core concept of the BERT model is to train a sophisticated bidirectional Transformer encoder that can effectively comprehend contextual information, resulting in robust word representations. Unlike traditional unidirectional language models, BERT harnesses both preceding and following context to enhance predictions, facilitating a superior grasp of semantics and contextual relationships. In the context of sentiment analysis, the BERT model processes input text by breaking it down into individual tokens. Initially, the text is tokenized, and special tokens are introduced to mark sentence boundaries. These tokens are then transformed into corresponding vector representations via an embedding layer. Conversely, the NSP task aims to determine whether two given sentences are consecutive in the original text. By pretraining on extensive corpora, The BERT model acquires comprehensive language representations that capture a holistic understanding of text, incorporating various linguistic features and contextual information. After the pretraining phase, BERT can be further adapted and customized for specific applications, for tasks like sentiment analysis, which involves analyzing the sentiment or emotion conveyed in text data, allowing it to excel in accurately gauging the sentiment expressed in text, named entity recognition, and more. This makes it a robust framework for analyzing sentiment in Twitter user text.

2.3. Implementation Details

In this section, the study provides details about the implementation of sentiment analysis using the BERT model for analyzing Twitter user text. including the learning rate, optimizer, loss function, evaluation metrics, and the number of training epochs. The implementation is based on the PyTorch framework.

2.3.1. Learning Rate and Optimizer

To ensure effective training of the BERT model. The study used approach is to utilize the Adam optimizer, which combines adaptive learning rates and momentum. The initial learning rate is 2×10^{-5}, and a warm-up strategy is employed where the learning rate gradually increases during the initial training steps. This approach facilitates faster convergence of the model and enhances training stability.

2.3.2. Loss Function

Choosing an appropriate loss function is crucial for effectively training sentiment analysis models. In the case of binary sentiment classification tasks, the study utilized the binary cross-entropy loss function. By using this loss function, the study was able to evaluate the deviation between the model's predictions and the true sentiments, leading to optimization and improvement of the performance of the sentiment analysis model.

2.3.3. Performance Metrics for Evaluation

To evaluate the effectiveness of the sentiment analysis model, the study employed various performance metrics, including accuracy, recall, and F1-score. These metrics were used to assess different aspects of the model's performance in correctly predicting sentiment labels. By considering multiple evaluation metrics, the study aimed to provide a comprehensive analysis of the model's overall performance and its ability to accurately classify sentiment.

2.3.4. Number of Training Epochs

In the implementation, the study perform hyperparameter tuning to determine the optimal number of training epochs based on the task-specific requirements and dataset characteristics. Finally, the BERT model gets the best model after training 20 times, and the RNN model gets the best model after training more than 50 times.

3. Results and discussion

During the experiment, two models were trained, RNN and BERT, on the collected dataset. It can be observed that after 20 training iterations, the accuracy of the RNN model reached only 85%, while the

DOI: 10.54254/2755-2721/52/20241380

BERT model achieved 96%. Figure 2 illustrates the variations in loss, accuracy, recall, and F1 score for both models during the training process from 1 to 20 iterations. It is evident that the BERT model consistently outperformed the RNN model throughout the entire training phase.

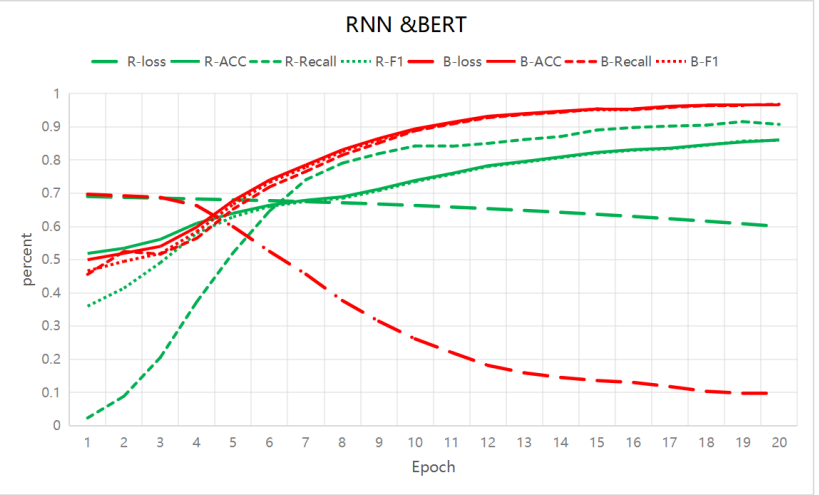

Figure 2. Trend chart of various performance indicators of RNN and BERT models (Photo/Picture credit: Original)

After training for over 50 iterations, the study obtained the final best performance metrics as shown in Figure 3 and Figure 4. In the RNN model, the F1 score reached 97.37%, the recall value reached 97.93%, and the accuracy reached 97.41%. In the BERT model, the F1 score reached 98.96%, the recall value reached 99.12%, and the accuracy reached 99.01%.

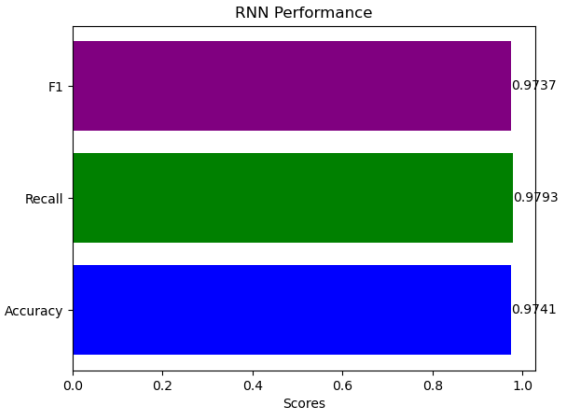

Figure 3. The best performance based on the RNN (Photo/Picture credit: Original)

DOI: 10.54254/2755-2721/52/20241380

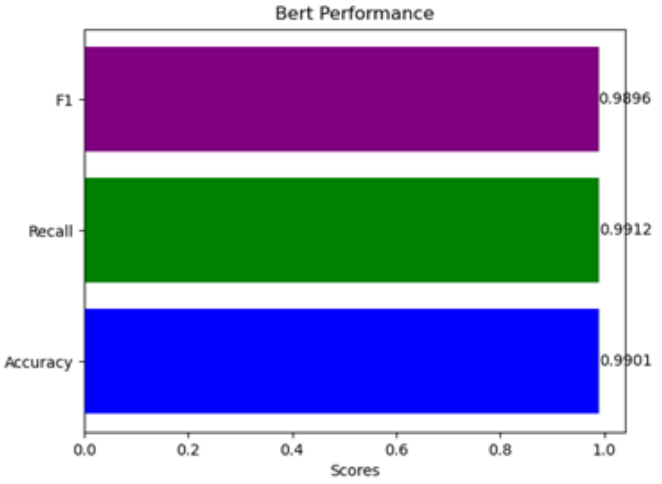

Figure 4. The best performance based on the BERT (Photo/Picture credit: Original)

Furthermore, in the experiment, this study employed ChatGPT for reannotation. The newly generated dataset was then incorporated into the preexisting models for training, yielding the results illustrated in Figure 5. Ultimately, this study observed an F1 score of 89.90%, a recall value of 83.33%, and an accuracy of 90.00%. Comparing these outcomes with the previous models, this study observed a decrease in accuracy of 9.8%. Upon further analysis, it was discovered that ChatGPT faced challenges in accurately discerning emotions related to certain human expressions or emoticons, leading to some errors.

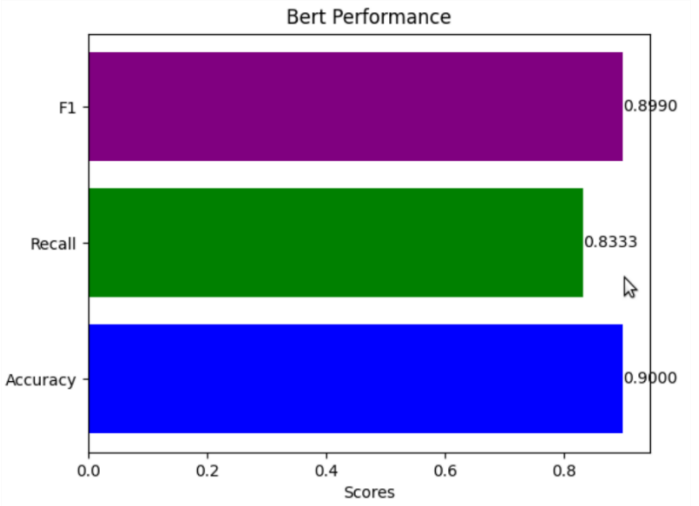

Figure 5. The best model trained by BERT after relabeling

Finally, after using two GPUs and adding the BERT model to the data parallel algorithm, it was found that the training speed was almost twice that of a single GPU. The performance of the BERT model increases by 65.05% after data parallelism and by about 147.98% when the data set is increased.

After further analysis, it was observed that ChatGPT faces challenges in accurately recognizing emotions related to certain human expressions or emoticons, leading to a decline in performance across various metrics. However, there are some limitations in this study. Due to the massive amount of data, the paper randomly selected a subset of the dataset for reannotation. As an improvement strategy, the dataset was randomly sliced, and ChatGPT was tasked with multiple rounds of reannotation. The newly

DOI: 10.54254/2755-2721/52/20241380

annotated data was then subjected to multiple training iterations using the same methodology to analyze the final results.

It is important to acknowledge that the random selection of data for reannotation might introduce biases or overlook certain patterns present in the overall dataset. Future studies could consider alternative approaches for data selection, such as stratified sampling, to ensure a more representative reannotation process.

Furthermore, although multiple training iterations were conducted with the reannotated data, it is worth noting that the impact of the reannotation process on the overall model performance needs to be thoroughly evaluated. It is possible that the iterative training approach could lead to overfitting or bias towards the reannotated samples. Therefore, additional experiments should be conducted to assess the robustness and generalizability of the improved model.

4. Conclusion

In addition to evaluating the BERT model's performance, this study also examines the capabilities of the chatgpt language model in sentiment analysis tasks. The findings suggest that chatgpt exhibits limitations in understanding and accurately capturing sentiment nuances. This highlights the need for further research and improvements in language models specifically designed for sentiment analysis. This research demonstrates the superiority of the BERT model over traditional RNN models in text sentiment analysis tasks. The incorporation of data parallelism and GPipe techniques significantly enhances training efficiency. Additionally, the study reveals the limitations of the chatgpt language model in sentiment analysis. Future research should focus on addressing these limitations and exploring novel approaches to improve sentiment analysis models.

References

[1] Saon G Tüske Z Bolanos D et al 2021 Advancing RNN transducer technology for speech recognition. ICASSP 2021-2021 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP) IEEE: 5654-5658

[2] Yadav S P Zaidi S Mishra A et al 2022 Survey on machine learning in speech emotion recognition and vision systems using a recurrent neural network (RNN) Archives of Computational Methods in Engineering 29(3): 1753-1770

[3] Prabha M I 2019 Srikanth G U. Survey of sentiment analysis using deep learning techniques 2019 1st international conference on innovations in information and communication technology (ICIICT). IEEE: 1-9

[4] Von Oswald J Niklasson E Randazzo E et al 2023 Transformers learn in-context by gradient descent International Conference on Machine Learning PMLR: 35151-35174

[5] Alaparthi S Mishra M 2020 Bidirectional Encoder Representations from Transformers (BERT): A sentiment analysis odyssey arXiv preprint arXiv:2007.01127

[6] Devlin J Chang M W Lee K et al 2018 Bert: Pre-training of deep bidirectional transformers for language understanding arXiv preprint arXiv:1810.04805

[7] Lin T Wang Y Liu X et al 2022 A survey of transformers AI Open

[8] Huang Y Cheng Y Bapna A et al 2019 Gpipe: Efficient training of giant neural networks using pipeline parallelism Advances in neural information processing systems 32

[9] Go A Bhayani R Huang L 2009 Twitter sentiment classification using distant supervision CS224N project report Stanford 1(12): 2009

[10] Kaggle 2017 Sentiment140 dataset with 1.6 million tweets https://www.kaggle.com/datasets/kazanova/sentiment140