Fine-Tuned Transformer Models for Question Answering

2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) | 979-8-3503-3509-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCCNT56998.2023.10307046

Sai Satvikh Lakkimsetty
Computer Science and Engineering
Gayatri Vidya Parishad College Of Engineering(Autonomous)
Vishakapatnam,India
saisatvikh@gmail.com

Sai Varshini Latchireddy
Computer Science and Engineering
Gayatri Vidya Parishad College Of Engineering(Autonomous)
Vishakapatnam,India
saivarshini2102@gmail.com

Sai Manikanta Lakkoju
Computer Science and Engineering
Gayatri Vidya Parishad College Of Engineering(Autonomous)
Vishakapatnam,India
20131a05b9@gmail.com

Gopalakrishna Reddy Manukonda
Computer Science and Engineering
Gayatri Vidya Parishad College Of Engineering(Autonomous)
Vishakapatnam,India
m.gopalkrishnareddy2@gmail.com

Dr. R.V.V.Murali Krishna
Information Technology
Gayatri Vidya Parishad College Of Engineering(Autonomous)
Vishakapatnam,India
rvvmuralikrishna@gvpce.ac.in

***Abstract*—Effective question-answering is crucial for applications involving natural language processing such as conversational artificial systems. The pre-trained BERT (Bidirectional Encoder Representations from Transformers) model achieved outstanding results on the Stanford Question Answering Dataset (SQuAD), a well-known benchmark for question answering tasks. The three different BERT models (BERT-base, BERT-large, DistilBERT) are pre-trained on general text corpus. These BERT models are trained to identify the right context in a paragraph to give answers.Further, fine-tuning the models with few-shot learning on domain related questions and answers improves the performance. Hence, our work proposes different strategies to improve the performance ofBERT models on tasks requiring domain-specific question-answering tasks using a customized SQuAD dataset. Our work highlights the significance of optimizing the BERT models on domain-specific data for enhanced performance inparticular tasks. The results of this study have implications for domain-specific knowledge in natural language processing. With a score of 0.9632, the BERT-Large model has the best accuracy out of all the BERT models tested. The performance of the BERT-Large model consistently beats that of the other models in several parameters, including precision, recall, and F1 score.**

Keywords—Transformers, Fine-Tuning, Few-shot learning, question-Answering

I. INTRODUCTION

Question-answering using machine learning is acrucial task in NLP[4] where the objective is to give a passageor a document . The discipline of Question-answering has advanced significantly with the introduction of pre-trained transformer models. These models are pre-trained natural language processing models developed by Google that uses a transformer basedarchitecture. In question-answering tasks, the BERT transformer model has attained cutting-edge performance. Our work focuses on optimizing different BERT models (BERT-Base[16], BERT-Large[14], DistilBERT[15]) for Question answering using a customized SQuAD dataset. SQuAD[2], a widely used benchmark dataset for Question answering, is made up of questions and answers based on a significant number of Wikipedia articles and general text corpus[3]. BERT Base contains 12 layers and 110 million parameters, making it appropriate for a wide variety of natural language processing jobs. BERT Large, with 24 layers and340 million parameters, is more powerful, and requires more computational resources. In DistilBERT, the number of parameters is reduced to 6 layers and 66 million parameters using knowledge distillation, making the model suitable for systems with limited computational resources. It is difficult to apply pre-trained BERT models to new domains and tasks, though BERT models' performance on certain tasks and domains can be enhanced by fine-tuning them on custom datasets.

Transfer learning is used to improve pre-trained BERT models performance on the custom dataset. To enhance the performance of different BERT models, fine-tuning methods and hyperparameters-tuning strategies are investigated. Fine Tuning is a process of increasing the models accuracy by modifying the training procedures such as alteringthe weights of the network and performing tasks such as regularization, stacking ,etc.[4]

The max-voting ensemble technique is used to increase the model's accuracy. Ensembling is a technique that combines the results of various models to produce results that are more effective compared to that of a single model. Max-voting is an effective method that involves selecting the class label that receives the maximum number of votes from the ensemble models. The model with the highest accuracy of the ones trained on our dataset is chosen using this technique. The accuracy of our model was greatly increased through the application of the ensembling technique with max-voting, which can be a useful tool in improving the performance of predictive models.

Optimization of BERT models significantly boost models' performance on the Question Answering task. RoBERTa (Robustly Optimised BERT Pre Training Approach)is developed for natural language processing problems. It is anextension of the well-known BERT model and enhances The remaining of the paper is organized as follows: Section II presents methodology involved in the

proposed work. The experimentation is described in Section III. The section IV describes the results obtained along with future scope. Finally,the paper is concluded in Section V.

II. METHODOLOGY

The techniques and different methods employed are discussed in this section. A custom SquAD Dataset in CSV format is used. By processing the dataset, a JSON file with necessary attributes is obtained. Upon fine-tuning the pre-trained model question answering task is being performed in an efficient manner. Figure 1 illustrates the methodology being followed.

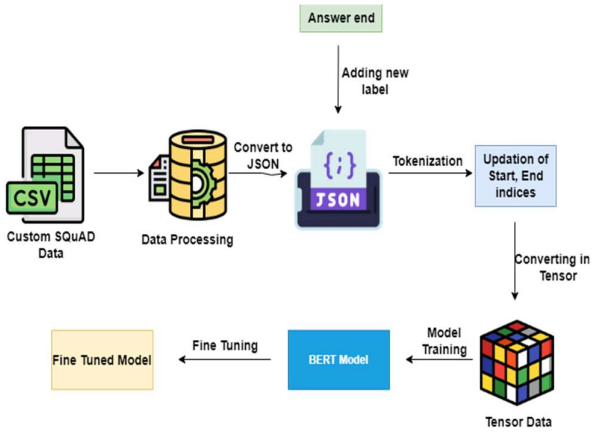


Figure 1: Workflow of the Proposed Question Answering System

The custom dataset which is being used is made by the ChatGPT language model[5]. The dataset consists of 500 passages and the corresponding questions, answers and answer start index. To guarantee quality and uniformity, each question is assigned to a unique question ID. The sample dataset is shown in Figure 2.

The raw data is first gathered, and any unnecessary orincorrect information is then removed. The questions and contexts are then taken out of the cleaned data and structured according to a standard format. In order to train the models, the final step involves padding the questions and contexts to make sure they are equal in length.

context	question	id	text	answer_start
Bowling is a sport that involves rolling a bal...	What is bowling?	77	a sport that involves rolling a ball down a la...	3.0
Bowling is a sport that involves rolling a bal...	What is a spare in bowling?	78	knocking down all remaining pins with the seco...	25.0
Snooker is a cue sport played on a table with ...	What is snooker?	79	a cue sport played on a table with pockets	3.0

Figure 2: Sample SQuAD dataset in CSV

A number of preprocessing operations as represented in Figure 3 are performed on the dataset to enhance data quality and consistency before models are trained. The questions, contexts, and answers are separated as individual columns. Using the answer start index and the length of the response, the response end index is calculated from the dataset provided.While tokenizing, the responses are padded to the maximum length and an answer end token is added to them.The available json dataset is tokenized and then the start indexand end index are updated in accordance with it.

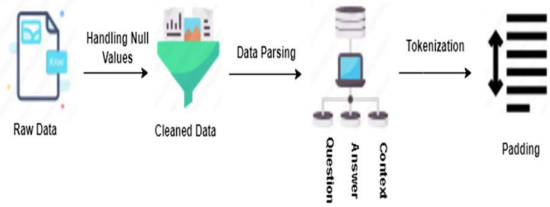


Figure 3: Data Preprocessing

Google Colab[19] and Tensorflow[6] are used for training and evaluation of various models. To access a single NVIDIA Tesla K80 GPU with 12GB of VRAM and 25GB of RAM, Google Colab's free version is used. For model training and inference, TensorFlow 2.5.0 and the Hugging Face Transformers library [7] are used.

The pre-trained BERT-base, BERT-large, and DistilBERT models from the Hugging Face Transformers library are imported to initialise our models[3]. The English Wikipedia corpus[8], the BookCorpus[9], and the OpenWebText corpus[10] are used to pre-train the BERT models.

These pre-trained models are trained on a custom SQuAD dataset which is converted into a TensorFlow dataset. The *fit()* function uses the legacy Adam optimizer[11] with a learning rate of 5e-5 over 10 iterations and a sparse categorical loss. A sequence length of maximum 256 tokens and a batch size of 16 is used. After each epoch, analysis of how well the fine-tuned BERT models performed on the validation set. The top model is chosen based on max-voting ensemble technique to improve the efficiency of the prediction .The details of the experiments performed are described in Section III.

III. EXPERIMENTATION

The experimentation section describes the significance of the various procedures used in the proposed work and compares itto those from earlier studies. This section also contains the study's limitations, recommendations for additional research, as well as potential applications of the findings.

The performance of different fine-tuned BERT models (DistilBERT, BERT-Base and BERT-Large) on a custom SQuAD dataset are compared. The fine-tuned BERT models are trained on features such as questions, contexts, answer start and answer end indices.

A. Evaluation Metrics

The various metrics such as F1 score, and latency are calculated on the test set to assess how well the fine tuned models are performing. To visualize the training and validation process, *matplotlib* [23] is used to plot the accuracy and loss graphs. The accuracy and loss of the models are evaluated.

- F1 Score: The formula for the F1 score is given by:

$$F1\ score = 2 \times \left(\frac{precision \times recall}{precision\ +\ recall}\right)$$

Where,

$$precision = \frac{TP}{(TP+FP)}$$

$$recall = \frac{TP}{(TP+FN)}$$

Where,

TP=True Positive

FP=False Positive

FN=False Negative

- Latency**:** It is described as the period of time betweenthe time a request is made and the time the responseis received in the context of computer networks and information systems.

$$Latency = (Model\ predictive\ end\ time - Model\ predictive\ start\ time)$$

- **Accuracy:** Accuracy is defined as the proportion of correctly answered questions to total questions in the system.

$$Accuracy = \frac{Total\ number\ of\ responses}{Total\ number\ of\ questions\ asked}$$

B. Challenges Faced During Emprical Analysis

There are various attempts made in order to improve the performance of the fine-tuned models. The relevant reasons for the ruined activities performed are mentioned below,

- Stacking Encoders of Different BERT Models:

Stacking the encoders of two separate BERT models (BERT-base and BERT-large) is done in orderto train a neural network for question answering tasks to enhance the model's performance. However, the model's accuracy is not growing despite adding more epochs and changing the learning rate.The chain rule is a calculus method that is used to quickly compute the gradients of the model parameters with respect to the loss function during the backpropagation process of training neural networks. It enables optimal deep neural network training by permitting gradients to traverse through the network's layers.

There could be problems with the chain rule when stacking encoders in a neural network, which would prevent the loss function from being updated. The chain rule is used in backpropagation. But when numerous encoders are stacked on top of one another, the chain rule might be violated, which could lead to mistakes in the gradient computation.

- Encoder-specific Analysis:

In our proposed work, the outcome of various BERT model layers are used to predict the ultimate result. It is noticed that the output is only predicted for the top 2 upper layers. The model's effectiveness consequently declined, and the overall performance of the model has decreased.

This problem arises because several layersof the BERT model's capture various kinds of data. Word associations and syntactical patterns are two examples of the more fundamental input properties that are captured by the lower layers. On the other hand, the top layers include the contextual and abstract characteristics. When the output of the upper levels is only considered, some of the crucial datathat the lower layers have captured can be overlooked, which leads to decreased efficiency and performance.

Therefore, while employing the BERT model for prediction tasks, it is crucial to take into account the output of the final encoder. By doing so,it is assured that every relevant piece of information is being used, which improves accuracy and efficiency.

- Training Other BERT Models Using TensorFlow:

Stacking the encoders of two separate BERT models (BERT-base and BERT-large) is done in orderto train a neural network for question answering tasks to enhance the model's performance. However, the model's accuracy is not growing despite adding more epochs and changing the learning rate.The chain rule is a calculus method that is used to quickly compute the gradients of the model parameters with respect to the loss function during the backpropagation process of training neural networks. It enables optimal deep neural network training by permitting gradients to traverse through the network's layers. Experiments are performed performed on ALBERT and RoBERTa. When attempting to train these models with TensorFlow, the computational resources available are exhausted. To overcome the issue TF training args are used, unlike the original BERT model, these models are not provided with the *model.fit()* and *compile()* methods.

The training procedure can be customized using the TF training args, which also enable fine- tuning of many parameters.However, compared to utilizing the conventional model.fit() and compile() approaches, this approach is more time consuming. It additionally requires more computational power and takes longer time to run. Even though ALBERT and

RoBERTa operate more effectively than the original BERT models, their installation and training creates special difficulties. Therefore, while choosing and applying various BERT models for certain tasks, it is crucial tocarefully analyze the consequences and practical concerns.

IV. RESULTS

In our proposed work, three distinct pre-trained language models (DistilBERT, BERT-Base, and BERT-Large) are used for question-answering tasks on a customized dataset. Various tests are performed to determine each models' performance and contrast them with one another.

The overall findings after fine-tuning of various pretrained BERT models are obtained. Accuracy, Latency, and F1 score are three common evaluation measures for question-answering tasks to assess the performance of the models. Error analysis revealed that the models had trouble answering queries requiring prior knowledge or common sense judgment, as well as queries including negation or ambiguity. Using the max voting ensemble method, the three BERT models' probabilities are calculated, compared, and the model with the highest probability is selected to make the prediction.

Table I. Evaluation results for the three models

Model	*Accuracy*	*F1_Score*	*Latency (ms)*
DistillBert	0.8701	0.7446	89.8484
Bert-Base	0.9264	0.7358	198.0945
Bert-Large	0.9632	0.7703	376.2423

The outcomes of our experiment are presented in Table I for each of the trained models. It is observed that the DistilBERT model had the lowest accuracy while the Bert-Large model had the highest accuracy and F1 scores. When compared to other BERT models, DistilBERT has an extremely low latency.

The F1 score, accuracy score, and latency score for the three models are shown in comparison in Figure 4. The BERT-Base model comes in second with the DistilBERT model having the lowest rating. The BERT-Large model has the greatest ratings.

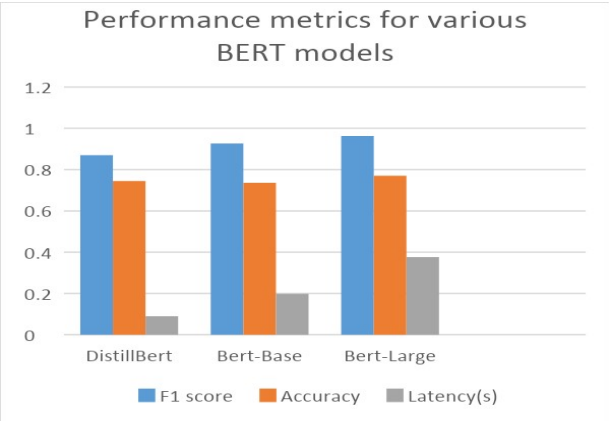


FIGURE 4: ACCURACY, LATENCY AND F1 SCORES COMPARISON FOR THE THREE MODELS

Table II. Evaluation results for the three models

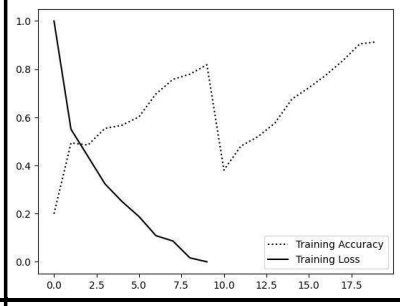
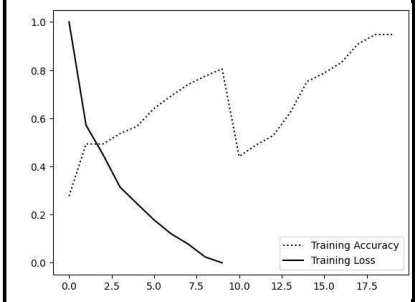
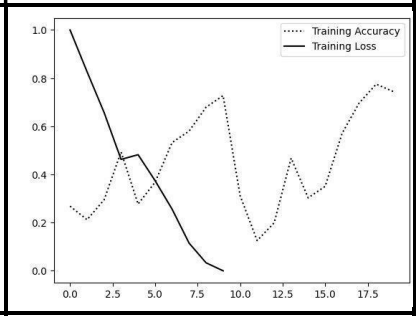
Model	*Accuracy and Loss*
DistilBERT	
Bert-Base	
Bert-Large	

Table 2 provides a summary of how well each fine-tuned BERT model performed based on the accuracy and loss metrics. With this one can compare the performance of each model and determine the best model.

A. *Future Scope:*

- Changes to Learning Rate and Batch Size

  Experimentations are carried out with different learning rates (1e-5, 2e-5, and 3e-5) and batch sizes (16, 32, and 64) to examine how the learning rates impact the models' accuracy throughout training. The best outcomes are obtained with a learning rate of 2e-5 and a batch size of 32.

- Modification of the optimizer function

  In addition to the Adam optimizer that is used for fine-tuning, other optimizer algorithms including Adagrad[21] and RMSprop[22] can also beevaluated. The Adam optimizer may produce the bestoutcomes.

- Analysis of Overfitting

  After a predetermined number of epochs, itis observed that the model is overfitting to the

training data. To solve this problem, early stopping is implemented which involves monitoring the validation loss and halting the training procedure when no more change in loss is observed. In the experiments performed using early stopping, the model's performance on the test data is increased and eliminates overfitting.

- Encoder-specific Analysis

  In the experiment, predicting the outcomes using any one encoder's output from the BERT model rather than the output from the entire model can also be analyzed. Every encoder in the model is crucial to forecast the outcome as evidenced by the fact that the models' performance is deprecated when output fromone encoder is used.

B. *Discussion:*

The various BERT models' performance has improved as a result of fine-tuning .F1 score, Accuracy, Latency are the three different metrics on which performance of the models is evaluated.To further evaluate the BERT models, max voting ensemble technique is used which reveals that BERT Large outperforms the BERT Base and DistilBERT models. As BERT large has more layers when compared to BERT Base and DistilBERT it captures the patterns present in the data well when compared to other models.

V. CONCLUSIONS

This study reveals that question-answering tasks on a customized SQuAD dataset can be accomplished effectively by pre-trained BERT models. Due to its capacity to identify complex relationships in text, the BERT-large model beat the other models.

The models were modified with the use of the unique SQuAD dataset, and they were assessed using a number of measures, such as F1 score, accuracy, and latency. On model performance, the effects of early halting and encoder-specific analysis were also assessed. The study was conducted with the highest ethical standards.

REFERENCES

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[2] SQuAD: 100000+ Questions for Machine Comprehension of Text. (n.d.). ACL Anthology from https://aclanthology.org/D16-1264/

[3] Reference corpora. (2023, January 19). CLARIN ERIC, from https://www.clarin.eu/resource-families/reference-corpora

[4] Fine-Tune Transformer Models For Question Answering On Custom Data. (n.d.). Towards Data Science. , from https://towardsdatascience.com/fine-tune-transformer-models-for-questi on-answering-on-custom-data-513eaac37a80

[5] OpenAI. (n.d.). Chatting with an AI model., from https://openai.com/blog/chatting-with-an-ai-model/

[6] Module:tf(2023,March24).TensorFlow.,fromhttps:// www.tnsorflow.org/api_docs/python/tf

[7] Models. (n.d.). Hugging Face., from https://huggingface.co/models

[8] English-Corpora: Wikipedia. (n.d.). English Corpora., from https://www.english-corpora.org/wiki/

[9] bookcorpus. (2022, June 28). TensorFlow., from https://www.tensorflow.org/datasets/community_catalog/hug gingface/bo okcorpus

[10] Dann, C., Mitchell, A., & Trump, D. (n.d.). openwebtext · Datasets at Hugging Face.Hugging Face., from https://huggingface.co/datasets/openwebtext

[11] keras.optimizers.Adam. (n.d.). Keras., from https://keras.io/api/optimizers/adam/

[12] ALBERT. (n.d.). Hugging Face., from https://huggingface.co/docs/transformers/model_doc/albert

[13] RoBERTa. (n.d.). Hugging Face., fromhttps://huggingface.co/docs/transformers/model_doc/roberta

[14] Devlin, J., Chang, W., Lee, K., & Toutanova, K. (2023, April 5). bert-large-uncased · Hugging Face. Hugging Face., from https://huggingface.co/bert-large-uncased

[15] DistilBERT. (n.d.). Hugging Face., fromhttps://huggingface.co/docs/transformers/model_doc/distilbert

[16] Devlin, J., Chang, W., Lee, K., & Toutanova, K. (2023, April 5). bert-base-uncased · Hugging Face. Hugging Face., from https://huggingface.co/bert-base-uncased

[17] The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning). (2018, December 3). Jay Alammar., from https://jalammar.github.io/illustrated-bert/

[18] The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time. (2018, June 27). Jay Alammar., from https://jalammar.github.io/illustrated-transformer

[19] (n.d.). Welcome To Colaboratory - Colaboratory., from https://colab.research.google.com/

[20] English-Corpora: Wikipedia. (n.d.). English Corpora., from https://www.english-corpora.org/wiki/Home. (n.d.). YouTube., from https://www.sciencedirect.com/science/article/abs/pii/S0048 9697150026 97

[21] keras.optimizers.Adagrad. (n.d.). Keras., fromhttps://keras.io/api/optimizers/adagrad/

[22] keras.optimizers.RMSprop (n.d.). Keras., fromhttps://keras.io/api/optimizers/RMSprop/

[23] (n.d.). Matplotlib — Visualization with Python. Retrieved April 30, 2023, from https://matplotlib.org/