Reviewing techniques to improve BERT Question Answering Model

Sourav Singh
Masters in Applied Business Analytics
Carleton University
Ontario, ON K1S 5B7
souravsingh3@cmail.carleton.ca

Abstract

BERT's existing question answering models are very expensive to train and maintain, so their practical use in the real world is difficult. The purpose of this paper is to analyse some of the existing methods for improving the BERT QA model. This study will provide a brief background on the application and suggest an improved approach to implementing the BERT model for a QA system. This project aims to analyze if BERT base QA model performance can be further improved using the combination of text pre-processing, adding an extra linear layer to the model and using the Adam optimizer. The generated question-answering BERT model is evaluated against SQuAD 1.1 dev benchmark data set. In particular, this project demonstrates how to create a custom model and ways to improve the performance of a model, this review also explores the challenges and issues with BERT for the QA system.

1 Introduction

There are various methods to structure the question-answering assignment. The most typical use is an extractive question answer in a limited setting. The SQuAD is a well-known dataset for QA, where the model chooses the word(s) that best describe the answer from a passage and a question. However, the majority of real-world uses for question-answering involve extremely extensive texts, such as an entire website or a large number of records in a database. To find the appropriate response, voice assistants like Google Home and Alexa search through a sizable collection of online documents.
In this paper, We are going to explore the BERT model for QA. The BERT (Devlin et al., 2018) (8) model has become a popular method of developing question-and-answer systems in the modern world. Generally, the model requires additional training (with a large relevant text corpus) in order to remain relevant when analyzing large complex documents (such as regulations, federal and institutional policies, and domain-specific documents). BERT has dominated the SQuAD 2.0 leaderboard since emerging and is already performing at a human level in Question Answering tasks. The BERT model has a number of advantages over other models, including the fact that it is effective for task-specific models and that it has been trained on a huge corpus, making it simpler for smaller, more precise NLP jobs. It is immediately usable and can be further adjusted. Additionally, the model is updated frequently, which contributes to its exceptional accuracy. More than 100 languages have pre-trained versions of the BERT model available. The fundamental drawback of the BERT model is that it takes a long time to train because it is large and has a lot of weights to update, despite the fact that it has several advantages. Therefore, the BERT model should be used by being aware of and comprehending what is best required in accordance with the demand.
Nevertheless, the predictions of the BERT model are still unable to resolve some natural language understanding issues (5). This paper will study some of the existing methods suggested for improving

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

the BERT model accuracy for the QA task, identify major challenges and suggest how we can utilize neural networks to improve the model accuracy

2 Related Work

To provide readers with a wider background, this paper will briefly examine prior related work. Looking at the past literature, it can be seen that **Text classification** and **question answers systems using BERT** has been the subject of excessive research.

2.1 QA Models

A model predicts the answer span in the input paragraph in a quality assurance task by producing the start and end positions of the response. A paragraph, which is a span of context, and a question are the inputs. Bi-Directional Attention Flow (BiDAF) (10) built on top of many bidirectional LSTMs to reach an EM/F1 score of 68.0/77.3 in the pre-BERT period on the SQuAD v1.1 data set. Many QA models developed in the post-BERT era are based on BERT or a BERT variation. On top of Transformers, BERT-based models are constructed (7). On the SQUAD, the original BERT study (1) achieves EM/F1 scores of 84.1/90.9 for the large-sized model and 80.8/88.5 for the small-sized model.

2.2 QA Models with text preprocessing techniques

It has been suggested that the BERT QA models can be improved through the use of techniques like **Definition Tokenization, Dependency Tokenization, Paragraph Splitting, Relevant Paragraph Ranking, and BERT Fine-tuning**. These techniques can reduce the content size and increase BERT accuracy on huge texts by **30–50 per cent** in terms of F1 score by using text processing techniques like paragraph splitting and relevant paragraph ranking (4).
Since it immediately deliver the most pertinent content, BERT with hand-chosen paragraphs sets the upper bound of our BERT performance. BERT using text processing techniques only loses 5–11 per cent of its F1 score accuracy in comparison to the top bound.

2.3 Improving QA models with Domain Specific Knowledge

K-AID is one of the recommended approaches that consists of a low-cost knowledge acquisition procedure for acquiring domain knowledge, a powerful knowledge infusion module for improving model performance, and a knowledge distillation component for shrinking the model size and deploying K-PLMs on resource-constrained devices (such as CPUs) for practical application. Importantly, the technique captures relational knowledge rather than entity knowledge, unlike the bulk of previous K-PLMs, which helps improve tasks essential to answering questions, such as text matching and categorization at the sentence level (QA). Their experimental results show that the approach can substantially improve sentence-level question-answering tasks and bring beneficial business value in industrial settings.(2)

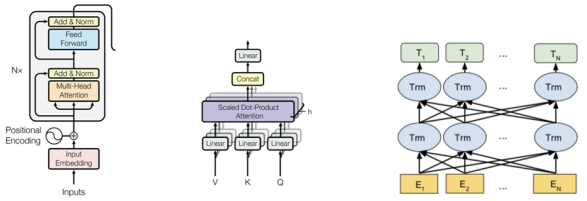
2.4 Huggingface Transformers library

Huggingface Transformers library contains a big collection of pre-trained models for many different tasks, including sentiment analysis, text summarization, paraphrasing, and of obviously question-answering. From the database of accessible models, users can pick a few potential question-answering models. As it turns out, several of these have already undergone refinement using the SQuAD dataset.(3) For reference, below are a few SQuAD fine-tuned models:

- distilbert-base-cased-distilled-squad
- ktrapeznikov/albert-xlarge-v2-squad-v2
- twmkn9/albert-base-v2-squad2
- bert-large-uncased-whole-word-masking-finetuned-squad
- mrm8488/bert-tiny-5-finetuned-squadv2

3 Approach

BERT Model Overview : Before we begin, it is important to understand the overview working of the existing BERT Bidirectional Encoder Representations from the Transformers model. BERT's model architecture is a multi-layer bidirectional Transformer encoder based on the original implementation described in Devlin et al.(2020) (8) BERT relies on several layers of Transformer blocks, note positional embeddings are not used in BERT. Each Transformer block consists of two sub-layers, a multi-head self-attention mechanism followed by a simple, position-wise fully connected feed-forward network Residual connections exist around each of the two sub-layers, and dropout, following after each sub-layer, provides layer normalization.



(a) Transformer Encoder (b) Multi-Head Attention (c) BERT Architecture

Figure 1: BERT Model Overview(11)

This section will discuss how I designed the model to function on data sets. This will also go over the training baseline model and the performance measures.

3.1 DistilROBERTA QA model

The RoBERTA-base model has been condensed into the DistilROBERTA QA model. Similar to DistilBERT, this model also goes through the training process. Since this model is case-sensitive, it can distinguish between the words English and English.(5). The model includes 12 heads, 768 dimensions, 6 layers, and 82M parameters in total (compared to 125M parameters for RoBERTa-base). DistilRoBERTa is typically twice as quick than Roberta-base.(3)
Although we can utilise the raw model for masked language modelling, its main purpose is to be refined on a later assignment. And this model is particularly meant to be improved on tasks like sequence classification, token classification, or question answering that require using the entire sentence (perhaps masked) to make conclusions.

3.2 Text Pre-processing and Adam Optimizer

For this project, I applied text pre-processing techniques to tokenize the input data and clean the dataset for any punctuations, and the obtained cleaned data was then fed to the DistilROBERTA with a linear layer for questions and paragraphs QA model for training. The approached QA model was trained in order to improve the performance accuracy of the base DistilROBERTA QA model.

3.2.1 Adam Optimizer

Gradient descent is the recommended method for optimising neural networks and many other machine learning methods. We used the Adam optimizer to improve the unique QA model. Adaptive Moment Estimation **(Adam)**(9) is a method that computes adaptive learning rates for each parameter. Adam maintains an exponentially decaying average of previous gradients, comparable to momentum, in addition to an exponentially decaying average of past squared gradients.
We can compute the decaying averages of past and past squared gradients as follows: The Adam

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

update rule is then obtained by using these to update parameters is computed as below: It was empirically demonstrated that Adam performs admirably in practise and outperforms other adaptive learning-method algorithms.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t.$$

3.3 Custom BERT QA Model Architecture

In this project, I have combined text pre-processing techniques and fed the processed data into a custom BERT using the DistilROBERTA QA model with linear layers for question embedding and paragraph embedding.

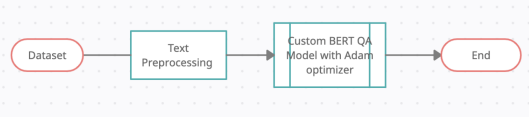


Figure 2: Question Answering model Workflow

4 Experiment

4.1 Dataset

I utilised the SQuAD 1.1 dev dataset for my experiment. I selected to investigate SQuAD 1.1 and extrapolate my findings for SQuAD 2.0 because the SQuAD 2.0 dataset is quite large and it was challenging to carry out the experiment on my PC as I waited long three days but my model only trained approximately 32% and then my system crashed.
The SQuAD 1.1 dev benchmark dataset includes more than 100,000 question-and-answer pairs on more than 500 articles. SQuAD (Stanford Question Answering Dataset) is a data set for reading comprehension. It consists of a list of questions by crowd workers on a set of Wikipedia articles. The answers to each of the questions is a segment of text, or span, from the corresponding Wikipedia reading passage. It's also possible that the query cannot be answered.(7)

4.2 Achieved Model Accuracy

4.2.1 F1 Score

In QA models, the F1 score is a popular indicator for categorization issues. When precision and memory are equally important to us, it makes sense. When it comes to QA models, it is calculated by comparing each word in the prediction to each word in the True Answer. The F1 score is based on the number of shared words between the prediction and the truth; recall is the ratio of shared words to the total number of words in the ground truth, and precision is the number of shared words to the total number of words in the prediction.

4.2.2 Model Accuracy: F1 Score

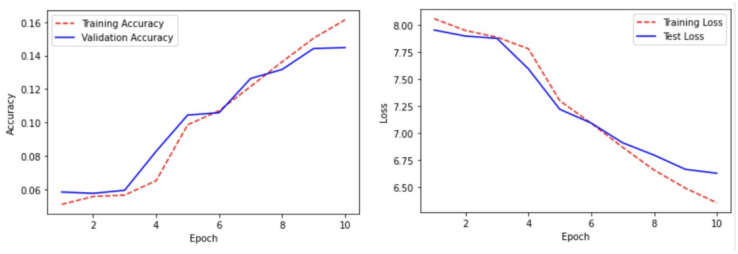


Figure 3: Model training accuracy and loss for 10 epochs

After training the model for 50 epochs on SQuAD data set, the overall accuracy achieved was very low with an F1 Score of mere 24.64. However, the accuracy could have been further increased if the model was allowed to train for more epochs.

4.2.3 Model Comparison chart for existing BERT Base model

Generally speaking the BERT model has already achieved a very high accuracy and improving its accuracy further requires a lot of training and fine tuning effort.
For this project and experimenting on BERT QA model, I chose **bert-large-uncased-whole-word-**

Model	F1	EM
ALBERT	86.97	78.39
BERT	81.51	71.82
DistilBERT	78.96	68.68
DistilROBERTA	87.74	80.39

Source: SQuAD 1.1 leaderboard

Figure 4: Accuracy comparison chart

masking-finetuned-squad base model. This model 24-layers, 1024 hidden dimensions, 16 attention heads and 336M parameters.
The model is pre-trained on 4 cloud TPUs in Pod configuration (16 TPU chips total) for one million steps with a batch size of 256 and fine tuned on the SQuAD dataset achieving an F1 Score of **93.15** and EM of **86.91**(3)

5 Results and Discussion

I performed three answer predictions scenarios to assess the resulting model. Here are the model predictions that were made after starting with a straightforward query and progressing to a complicated, then impossible-to-answer, scenario.

5.1 Basic Scenario

```
context = "Hi! My name is Sourav and I am 30 years old. I live in Ottawa Canada"

queries = ["What is my name?",
           "Where does Sourav live?",
           "What is the age of Sourav?"
          ]
answers = ["Sourav",
           "Ottawa Canada",
           "30"
          ]

for q,a in zip(queries,answers):
  give_an_answer(context,q,a)
```

```
Question: What is my name?
Prediction: sourav
True Answer: Sourav
EM: 1
F1: 1.0


Question: Where does Sourav live?
Prediction: ottawa canada
True Answer: Ottawa Canada
EM: 1
F1: 1.0


Question: What is the age of Sourav?
Prediction: 30
True Answer: 30
EM: 1
F1: 1.0
```

Figure 5: Model answer prediction for Basic Scenario

The Model was given a context and asked a very simple and easy to answer question. The model performs outstanding when predicting the answers of those easy to answer questions with 100 percent of accuracy.

5.2 Complex Scenario

When asked the model to look up for a complex relation, model didn't able to compute the exact result, the logic of $X > YandZ > X => Z > XandY$ is not captured at all.

```
context = "team X beated team Y in this game, but team Z"

queries = ["which team is the first place?"
           ]
answers = ["team Z"
           ]

for q,a in zip(queries,answers):
  give_an_answer(context,q,a)
```

```
Question: which team is the first place?
Prediction: team x
True Answer: team Z
EM: 0
F1: 0.5
```

Figure 6: Model answer prediction for Complex Scenario

5.3 Impossible to answer Scenario

When asked the model, a very easy but complex question. The model fails to compute the exact answer. Though the model accuracy is almost human level. It fails to answers what seems very easy for a human to answer. This explains that even though the model accuracy is quite high it is impossible for the model to carry out logical computation

```
context = "Sourav drink two cup of tea in morning and one in night"
queries = ["How many cup of tea does Sourav drink in a day?",
           "How many cup of tea does Sourav drink in night?"
          ]
answers = ["three",
           "one"
          ]

for q,a in zip(queries,answers):
  give_an_answer(context,q,a)
```

```
Question: How many cup of tea does Sourav drink in a day?
Prediction: two
True Answer: three
EM: 0
F1: 0


Question: How many cup of tea does Sourav drink in night?
Prediction: one
True Answer: one
EM: 1
F1: 1.0
```

Figure 7: Model answer prediction for impossible scenario

6 Conclusion

I am aware that this model is merely fair and not the best one currently available in research. But I believe that the experiment performed on scenarios still demonstrates the potential and highlight the limitations of the BERT QA model, as well as what the existing NLP model can accomplish and what has to be improved going forward.
In experiment 1, it was demonstrated that the QA model could recognise basic synonyms and extract unique information about query keywords. Finally, the model failed to grasp "numbers" and had no mathematical aptitude at all, which can be shown in the last experiment. The model is also weak at "reasoning" if the relationship is moderately complicated, as can be observed in experiment 2.

7 Future Work

The model was only trained for 50 epochs, and that too in a constrained environment using the older dataset, due to the CPU's limitations. In future, I want to add dense layers to my custom model and integrate it with text pre-processing methods already in use to examine how well it performs. Further investigation into how these models function internally when making predictions about the outcomes using various interpretation techniques would be intriguing and could reveal how they might be employed to provide solutions to challenging and impossible problems. There are several applications that can be built on top of the current BERT QA architecture if these restrictions are addressed.

References

[1] blackZhang, Y. and Li, J. “The Death of Feature Engineering? —— BERT with Liguistic Features on SQuAD 2.0,” *Stanford University*. black

[2] black Fu, S., Feng-Lin, L., Wang, R., Chen, Q., Cheng, X., and Zhang, J. (2021). "K-AID: Enhancing Pre-trained Language Models with Domain Knowledge for Question Answering. "*arXiv.org* black

[3] blackhttps://huggingface.co/ black

[4] blackLiao, C., Maniar, T., Sravanajyothi, N., and Sharma, A. (2020). Techniques to Improve QA Accuracy with Transformer-based models on Large Complex Documents. *arXiv.org*.black

[5] black Thambi, S. V., and ReghuRaj, P. C. (2022). Towards Improving the Performance of Question Answering System using Knowledge Graph - A Survey. *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 672–679. https://doi.org/10.1109/ICAIS53314.2022.9742802 black

[6] black Squad2.0 The Stanford Question Answering Dataset. Available at: https://rajpurkar.github.io/SQuAD-explorer/ black

[7] black’Squad1.1 dev dataset’ *DeepAI*. Available at: https://deepai.org/dataset/squad1-1-dev.black

[8] blackDevlin, J. et al. (no date) “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,”. *arXiv.org*.black

[9] blackKingma, D. P., and Ba, J. L. (2015). Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations*, 1–13. black

[10] blackDing, B. and Wang, Y. “Improving Bi-Directional Attention Flow for Machine Comprehension,” *Stanford University*black

[11] blackVaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need. *arXiv.org*.black