CPSC 8430
Deep Learning Homework-3
Avinash Komatineni

GithubLink: https://github.com/avinash-komatineni/-DeepLearning_HW3

Bonus Attemptted:
1. Automatic learning rate decay
2. Changing doc stride(Pre Processing)
3. Automatic mixed precision
4. Gradient accumulation
5. Post processing
6. Other pretraining models

Introduction:
Extractive question answering is a type of system that extracts the answer from given context. BERT is a powerful language model that can be used to build such systems. The SQuAD dataset is popular dataset for extractive question answering.

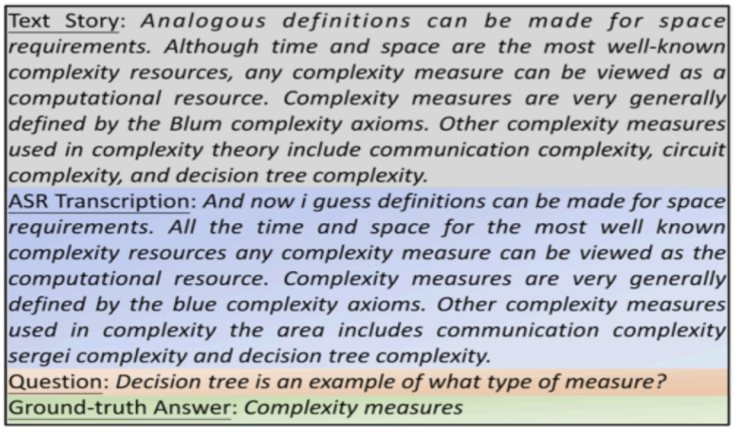
The BERT-based approach to extractive question answering involves fine-tuning the pre-trained BERT model on the SQuAD dataset. During the inference steps, the model takes the context and the question as input and output the start and end positions of the answer. These positions can be used to extract the answer from the context and provide it as the output. This approach has shown very good results on the SQuAD dataset and has been used in various real-world applications.

Dataset:
SpokenSQuAD is a unique dataset comprising spoken documents, text input questions, and answers that are always a span within the spoken document. This dataset was created by converting the original SQuAD dataset's articles into spoken form using Google text-to-speech system and generating corresponding ASR transcriptions using CMU Sphinx.

During the creation of the SpokenSQuAD dataset, the training and testing sets were generated from the SQuAD training and development sets, respectively. However, if a question's answer was not present in the associated article's ASR transcriptions, that question-answer pair was eliminated from the dataset.

The purpose of creating SpokenSQuAD was to provide dataset for training and evaluating question answering models that can handle spoken language input. This dataset can be used to develop and test models that can extract answers from spoken documents using text-based questions.

We have taken SQuAD Dataset, it consists of 37,111 question answer pairs as training set(Word Error Rate 22.77%) and 5,351 as testing set (Word Error Rate 22.73%).



Requirements:

Python
Torch
SQUAD Dataset
Scipy

Preprocess:

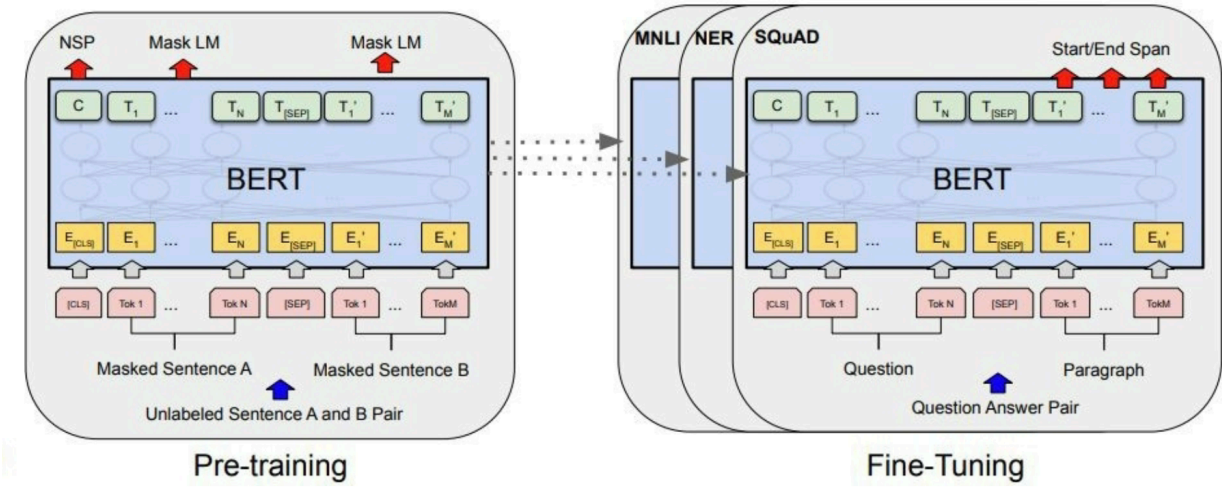
The dataset is provided in JSON format, which can be difficult. Therefore, it is often necessary to convert the SQuAD data from JSON to a more easily readable format, such as CSV.

Converting SQuAD data from JSON to CSV involves extracting the relevant information from the JSON data and writing it to a new CSV file in a comma-separated format. This is done using, the JSON data can be read using a JSON library and the CSV file can be created and written using a CSV library. The resulting CSV file can be easily imported into tools for further analysis and processing.

Model:

The BERT model is a pre-trained language model that uses a transformer architecture to learn contextualized word embeddings. Pretraining has training the model on a large amount of text data, such as Wikipedia or BookCorpus, to learn general language understanding, while fine-tuning involves training the model on a specific task, such as extractive QA.

To fine-tune the BERT model for extractive QA, the SQuAD dataset is used, which consists of context paragraphs and questions with corresponding answer spans. The BERT model is trained predict the start and end indices of the answer within the context paragraph. During fine-tuning, the entire model is fine-tuned on the SQuAD training set, which consists of over 100,000 question-answer pairs.



Long paragraphs can be an issue in extractive question answering because they contain a large amount of information that may not be relevant to the answer to a particular question. This makes it computationally expensive, time-consuming for the model to scan the entire paragraph to find the answer, leading to reduced performance. Additionally, long paragraphs may contain multiple possible answers to a question, which can make it challenging for the model to determine the most appropriate answer. Techniques such as dividing long paragraphs into smaller segments using a sliding window approach or using a hierarchical approach to first identify the relevant paragraphs in the context before extracting the answer span have been proposed to address this issue.

The fine-tuned BERT model can be used to predict the answer to new questions within a given context. The model is evaluated on a separate testing set to assess its performance on unseen data.

Results:

Bonus

Automatic learning rate decay: It is a technique used to adjust the learning rate during training to improve the performance of the model in extractive QA tasks.

Changing doc stride (Pre-Processing): It is a technique used in extractive QA to handle long paragraphs by splitting them into smaller segments to improve the model's performance.

Automatic mixed precision: It is a technique used to improve the performance and reduce the memory usage of the model in extractive QA by using lower precision data types for certain computations.

Gradient accumulation: It is a technique used to improve the training of the model in extractive QA by accumulating gradients from multiple mini batches before updating the model weights.

Post processing: It is a technique used in extractive QA to refine the output of the model after the inference to improve the accuracy of the predictions.

Other pretraining models: There are several pre-training models that have been developed to improve the performance of the BERT model for extractive QA tasks, such as RoBERTa, ALBERT, and DistilBERT. Here I used DistilBERT model.

After training and testing here are the results

```
bert_model = bert_model.to(device)
for epoch in range(num_epochs):
    train_loss = model_train(bert_model, train_data_loader, optimizer, device, accumulation_steps=2)
    test_loss = model_test(bert_model, test_data_loader, optimizer, device)

    print(f'Epoch {epoch+1} ---> train loss {train_loss} ---> test loss {test_loss}')

    # Update learning rate
    scheduler.step(test_loss)
```

```
Epoch 1 ---> train loss 5.3849138663681835 ---> test loss 5.4383845165349625
Epoch 2 ---> train loss 5.307152661587942 ---> test loss 5.3353072385260525
Epoch 3 ---> train loss 5.089821841312972 ---> test loss 5.265667626319623
```

```
distilbert_model = distilbert_model.to(device)

for epoch in range(num_epochs):
    train_loss = model_train(distilbert_model, train_data_loader, optimizer, device, accumulation_steps=2)
    test_loss = model_test(distilbert_model, test_data_loader, optimizer, device)
    print(f'Epoch {epoch+1} ---> train loss {train_loss} ---> test loss {test_loss}')

    scheduler.step(test_loss)
```

```
Epoch 1 ---> train loss 5.510384187986421 ---> test loss 5.601940179263173
Epoch 2 ---> train loss 5.566739558144809 ---> test loss 5.58954277665804
Epoch 3 ---> train loss 5.526833188829555 ---> test loss 5.491155148978012
```

After comparing the train loss and test loss BERT model is performing better than DistilBERT model

Predictions:

[43]:
```
context = '''this main building and the library collection was entirely destroyed by a fire in april eighteen
             seventy nine and the school closed immediately and students were sent home. the university founder
             f r. soaring and the president at the time the rent. william corby immediately plan for the
             rebuilding of the structures that have housed virtually the entire university.
             construction was started on the seventeenth of may and by the incredible zeal of a administrator
             and workers the building was completed before the fall semester of eighteen seventy nine.
             the library collection was also rebuilt in spain housed in the new main building for years
             afterwards. around the time of the fire the music hall was opened. eventually becoming known as
             washington hall a hosted placing musical act put on by the school. by eighteen eighty as
             science program was established at the university and a science hall today lafferty in student
             center was built in eighteen eighty three. the hall housed multiple classrooms in science labs
             needed for early research at the university.'''

question = "What was the music hall at Notre Dame called?"

answer = predict_answer(bert_model, tokenizer, context, question)

print(answer)
```

```
washington
```

[44]:
```
context = "The quick brown fox jumps over the lazy dog."
question = "What does the fox jump over?"
answer = predict_answer(bert_model, tokenizer, context, question)
print(answer)
```

```
the lazy dog
```

Both BERT and DistilBERT model gave almost similar predictions.