Fine-Tuning BERT for Question Answering on SQuAD Dataset

Shubham Kose (skose@purdue.edu)

Introduction

In terms of natural language processing (NLP), Google's Bidirectional Encoder Representations from Transformers (BERT) represents a significant advancement. BERT, which Google developed in 2018, transforms language understanding by allowing it to capture complex nuances and semantics by bidirectionally contextualizing words within a sentence. Its revolutionary architecture and training approach have brought it to the forefront of NLP research, enabling applications in a wide range of industries, including virtual assistants, search engines, healthcare, and finance.

Fine tuning BERT

Fine-tuning BERT for downstream tasks, such as Question Answering (QA), typically involves several key steps. Firstly, the pre-trained BERT model, with its multiple layers and attention mechanisms, serves as a feature extractor, encoding input text into contextualized embeddings. Next, a task-specific layer, often a neural network, is added on top of BERT to tailor it to the QA task. Fine-tuning BERT for QA tasks requires careful consideration of hyperparameters such as number of epochs we trained the model for, in this case we trained the final model for 21 epochs which is the model with best producing results.

Train output for the first 5 epochs

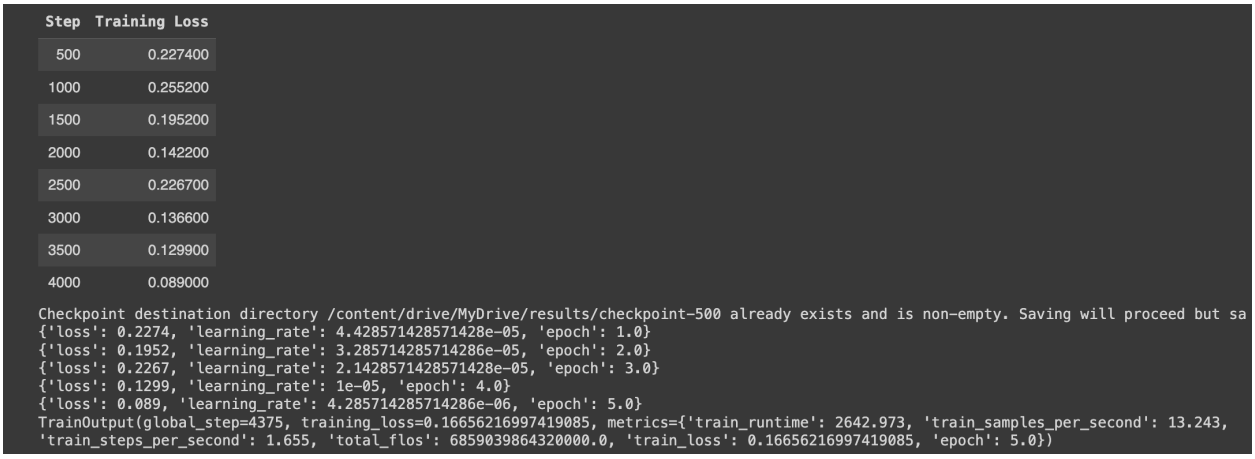
Code

```
training_args = TrainingArguments(
    output_dir='./results',  # output directory
    num_train_epochs=5,  # total number of training epochs
```

```
    per_device_train_batch_size=8,  # batch size per device d
uring training
    per_device_eval_batch_size=8,  # batch size for evaluatio
n
    weight_decay=0.01,  # strength of weight decay
    logging_dir='./logs',  # directory for storing logs
)

train_dataset = Dataset.from_pandas(pd.DataFrame(train_proces
sed))
eval_dataset = Dataset.from_pandas(pd.DataFrame(eval_processe
d))
test_dataset = Dataset.from_pandas(pd.DataFrame(test_processe
d))


# Custom callback class
class EpochLoggingCallback(TrainerCallback):
    def on_epoch_end(self, args, state, control, **kwargs):
        epoch = state.epoch
        logs = state.log_history[-1]  # Get the latest logged
metrics

        # Extract the loss and learning rate from the Trainer
        loss = logs.get("loss")
        learning_rate = logs.get("learning_rate")

        # Print the loss, learning rate, and epoch informatio
n
        print(f"{{'loss': {loss}, 'learning_rate': {learning_
rate}, 'epoch': {epoch}}}")

# Initialize Trainer
trainer = Trainer(
    model=model,
    args=training_args,
```

```
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
)

# Attach the custom logging callback to the Trainer to print
after each epoch
trainer.add_callback(EpochLoggingCallback())

# Train the model
trainer.train()
```

Output:



Step	Training Loss
500	0.227400
1000	0.255200
1500	0.195200
2000	0.142200
2500	0.226700
3000	0.136600
3500	0.129900
4000	0.089000

```
Checkpoint destination directory /content/drive/MyDrive/results/checkpoint-500 already exists and is non-empty. Saving will proceed but sa
{'loss': 0.2274, 'learning_rate': 4.428571428571428e-05, 'epoch': 1.0}
{'loss': 0.1952, 'learning_rate': 3.285714285714286e-05, 'epoch': 2.0}
{'loss': 0.2267, 'learning_rate': 2.1428571428571428e-05, 'epoch': 3.0}
{'loss': 0.1299, 'learning_rate': 1e-05, 'epoch': 4.0}
{'loss': 0.089, 'learning_rate': 4.285714285714286e-06, 'epoch': 5.0}
TrainOutput(global_step=4375, training_loss=0.16656216997419085, metrics={'train_runtime': 2642.973, 'train_samples_per_second': 13.243,
'train_steps_per_second': 1.655, 'total_flos': 6859039864320000.0, 'train_loss': 0.16656216997419085, 'epoch': 5.0})
```

```
{'loss': 0.2274, 'learning_rate': 4.428571428571428e-05, 'epo
ch': 1.0}
{'loss': 0.1952, 'learning_rate': 3.285714285714286e-05, 'epo
ch': 2.0}
{'loss': 0.2267, 'learning_rate': 2.1428571428571428e-05, 'ep
och': 3.0}
{'loss': 0.1299, 'learning_rate': 1e-05, 'epoch': 4.0}
{'loss': 0.089, 'learning_rate': 4.285714285714286e-06, 'epoc
h': 5.0}

TrainOutput(global_step=4375, training_loss=0.166562169974190
```

```
85, metrics={'train_runtime': 2642.973, 'train_samples_per_se
cond': 13.243, 'train_steps_per_second': 1.655, 'total_flos':
6859039864320000.0, 'train_loss': 0.16656216997419085, 'epoc
h': 5.0})
```

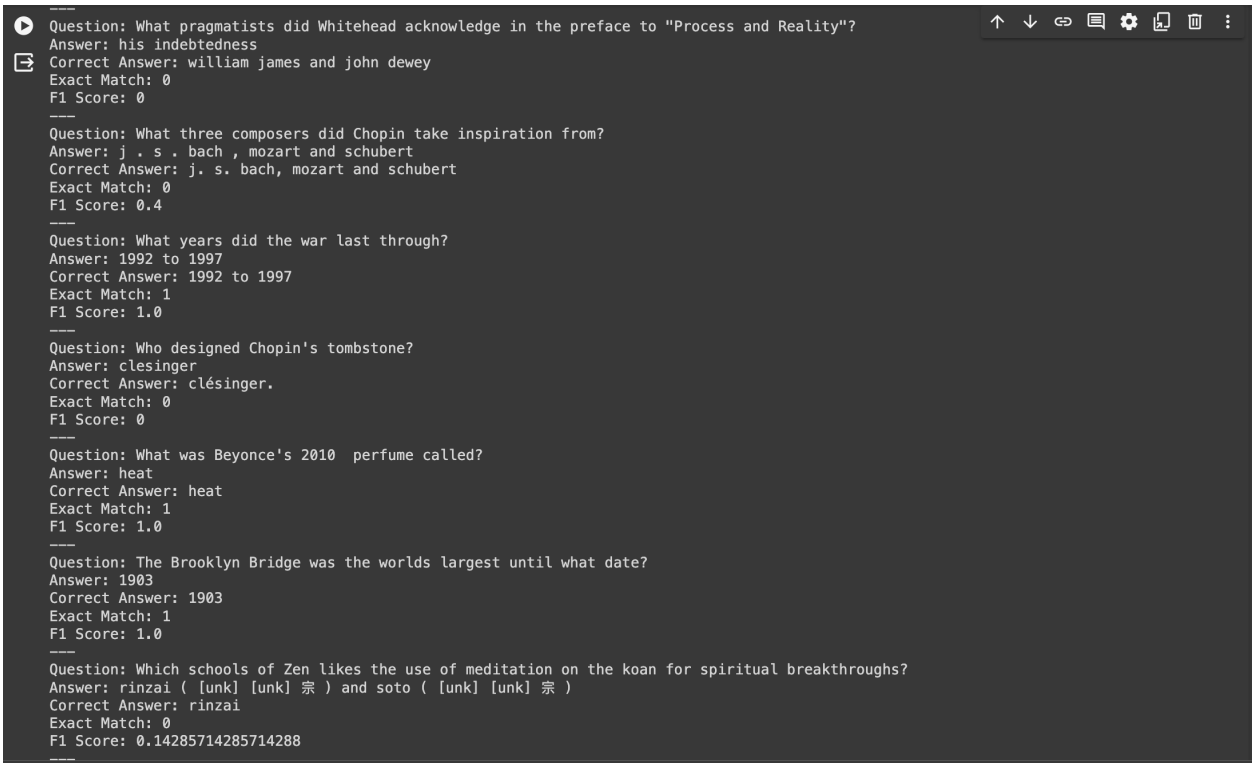
Qualitative Evaluation:

Upon qualitatively inspecting 10-20 answers generated by the model, it appears that the model's outputs vary in quality. Some answers may closely match the correct answers, demonstrating the model's capability to understand and extract relevant information from the context. These answers contribute positively to the higher F1 scores observed. However, there are also instances where the answers generated by the model are inaccurate or irrelevant, resulting in lower EM scores.

It is important to note that while the model exhibits satisfactory performance on certain questions, it struggles with others, particularly those involving nuanced language, complex contexts, spaced answers, words with bigger lengths, or ambiguous queries.

Overall, while the model's performance is not perfect, the obtained results suggest that it possesses some degree of understanding and proficiency in answering questions based on the provided context. Further refinement and experimentation may lead to enhancements in performance and better alignment with the desired outcomes.

Outputs:




```
Question: What type of activity did early settlers use to get
food that didn't involve farming?
Answer: fishing
Correct Answer: fishing
Exact Match: 1
F1 Score: 1.0
---
Question: Certain dogs are bred to help fishermen with what?
Answer: nets
Correct Answer: nets
Exact Match: 1
F1 Score: 1.0
---
Question: How many albums does Kanye have on the "500 Greates
t Albums of All Time" list?
Answer: 32 million
Correct Answer: 3
```

```
Exact Match: 0
F1 Score: 0
---
Question: Who could be made vulnerable by the Gasemtschulen?
Answer: bright working class students
Correct Answer: bright working class students
Exact Match: 1
F1 Score: 1.0
---
Question: How many magazines can call NYC home?
Answer: 350
Correct Answer: 350
Exact Match: 1
F1 Score: 1.0
---
Question: How many schools collapsed in Mianyang City?
Answer: seven
Correct Answer: seven
Exact Match: 1
F1 Score: 1.0
---
Question: Which character in the film, Epic, was voiced by Be
yoncé?
Answer: queen tara
Correct Answer: Queen Tara
Exact Match: 0
F1 Score: 0
---
Question: How many helicopter were to be provided by the civi
l aviation industry?
Answer: 30
Correct Answer: 30
Exact Match: 1
F1 Score: 1.0
---
Question: How many new infections of resistant TB are reporte
```

```
d per year?
Answer: half a million
Correct Answer: half a million
Exact Match: 1
F1 Score: 1.0
---
Question: What has the Polish government not allowed to find
true cause of death?
Answer: dna testing
Correct Answer: DNA testing
Exact Match: 0
F1 Score: 0.5
---
Question: Which year did PETA spark controversy with Beyonce?
Answer: 2006
Correct Answer: 2006
Exact Match: 1
F1 Score: 1.0
---
Question: What pragmatists did Whitehead acknowledge in the p
reface to "Process and Reality"?
Answer: his indebtedness
Correct Answer: william james and john dewey
Exact Match: 0
F1 Score: 0
---
Question: What three composers did Chopin take inspiration fr
om?
Answer: j . s . bach , mozart and schubert
Correct Answer: j. s. bach, mozart and schubert
Exact Match: 0
F1 Score: 0.4
---
Question: What years did the war last through?
Answer: 1992 to 1997
Correct Answer: 1992 to 1997
```

```
Exact Match: 1
F1 Score: 1.0
---
Question: Who designed Chopin's tombstone?
Answer: clesinger
Correct Answer: clésinger.
Exact Match: 0
F1 Score: 0
---
Question: What was Beyonce's 2010  perfume called?
Answer: heat
Correct Answer: heat
Exact Match: 1
F1 Score: 1.0
---
Question: The Brooklyn Bridge was the worlds largest until wh
at date?
Answer: 1903
Correct Answer: 1903
Exact Match: 1
F1 Score: 1.0
---
Question: Which schools of Zen likes the use of meditation on
the koan for spiritual breakthroughs?
Answer: rinzai ( [unk] [unk] 宗 ) and soto ( [unk] [unk] 宗 )
Correct Answer: rinzai
Exact Match: 0
F1 Score: 0.14285714285714288
---
Question: Because of the earthquake, how many people did not
have housing?
Answer: 5 million
Correct Answer: at least 5 million
Exact Match: 0
F1 Score: 0.6666666666666666
---
```

```
Question: In what year did Chopin become a French citizen?
Answer: 1835
Correct Answer: 1835
Exact Match: 1
F1 Score: 1.0
---
```

Quantitative Evaluation metrics:

Based on the obtained results, the average exact match (EM) score is approximately 0.455, and the median EM score is 0.0. Similarly, the average F1 score is approximately 0.612, and the median F1 score is 0.8. These scores indicate the performance of the model in correctly identifying the answers, which tells us that however the model doesn't perform the best, it still gives pretty decent results.

```
Average EM: 0.455
Median EM: 0.0
Average F1 Score: 0.6122484239379669
Median F1 Score: 0.8
```

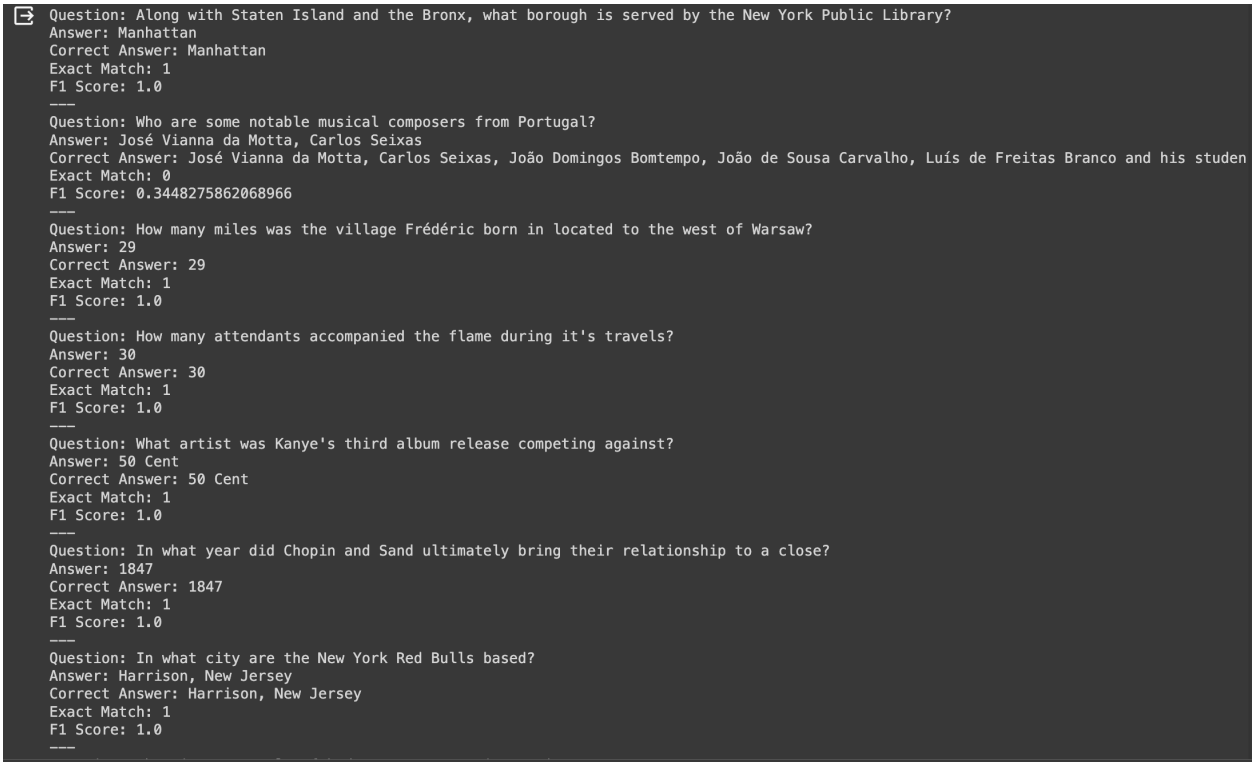
Average EM: 0.455
Median EM: 0.0
Average F1 Score: 0.6122484239379669
Median F1 Score: 0.8

Comparison of our fine tuned model with distilbert-base-cased-distilled-squad model from Hugging Face

Qualitative Evaluation for distilbert-base-cased-distilled-squad model:

As we can see below the outputs from the distilbert-base-cased-distilled-squad model are very good and most answers are answered correctly.

Outputs:




```
Question: Along with Staten Island and the Bronx, what borough is served by the New York Public Library?
Answer: Manhattan
Correct Answer: Manhattan
Exact Match: 1
F1 Score: 1.0
---
Question: Who are some notable musical composers from Portugal?
Answer: José Vianna da Motta, Carlos Seixas
Correct Answer: José Vianna da Motta, Carlos Seixas, João Domingos Bomtempo, João de Sousa Carvalho, Luís de Freitas Branco and his student Joly Braga Santos
```

```
Exact Match: 0
F1 Score: 0.3448275862068966
---
Question: How many miles was the village Frédéric born in loc
ated to the west of Warsaw?
Answer: 29
Correct Answer: 29
Exact Match: 1
F1 Score: 1.0
---
Question: How many attendants accompanied the flame during i
t's travels?
Answer: 30
Correct Answer: 30
Exact Match: 1
F1 Score: 1.0
---
Question: What artist was Kanye's third album release competi
ng against?
Answer: 50 Cent
Correct Answer: 50 Cent
Exact Match: 1
F1 Score: 1.0
---
Question: In what year did Chopin and Sand ultimately bring t
heir relationship to a close?
Answer: 1847
Correct Answer: 1847
Exact Match: 1
F1 Score: 1.0
---
Question: In what city are the New York Red Bulls based?
Answer: Harrison, New Jersey
Correct Answer: Harrison, New Jersey
Exact Match: 1
F1 Score: 1.0
```

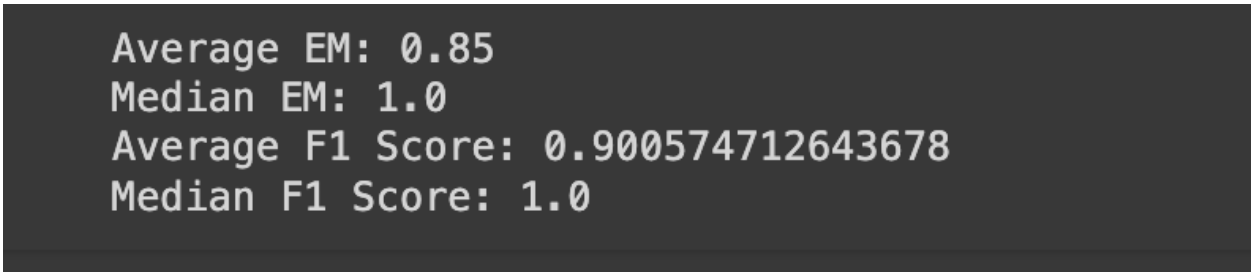
```
---
Question: What is an example of bad treatment causing resista
nce?
Answer: penicillin and erythromycin
Correct Answer: overuse of antibiotics
Exact Match: 0
F1 Score: 0
---
Question: Their third album, Survivor, sold how many during i
ts first week?
Answer: 663,000
Correct Answer: 663,000 copies
Exact Match: 0
F1 Score: 0.6666666666666666
---
Question: How many different breeds are there?
Answer: hundreds
Correct Answer: hundreds
Exact Match: 1
F1 Score: 1.0
---
Question: Unequal crossing over can create what type of repet
itive DNA?
Answer: Tandem repeats
Correct Answer: Tandem repeats
Exact Match: 1
F1 Score: 1.0
---
Question: What city saw the largest growth?
Answer: Kalispell
Correct Answer: Kalispell
Exact Match: 1
F1 Score: 1.0
---
Question: Beyonce was coached for her Spanish songs by which
American?
```

```
Answer: Rudy Perez
Correct Answer: Rudy Perez
Exact Match: 1
F1 Score: 1.0
---
Question: In which season did Coca-Cola become a sponsor of A
merican Idol?
Answer: season one
Correct Answer: season one
Exact Match: 1
F1 Score: 1.0
---
Question: Who suggested that Chopin's preludes were not inten
ded to be played as a group?
Answer: Kenneth Hamilton
Correct Answer: Kenneth Hamilton
Exact Match: 1
F1 Score: 1.0
---
Question: Who directed The Living Daylights and Licence to Ki
ll?
Answer: John Glen
Correct Answer: John Glen
Exact Match: 1
F1 Score: 1.0
---
Question: What are three games, in addition to Brick, which h
ave been included with the iPod?
Answer: Parachute, Solitaire, and Music Quiz
Correct Answer: Parachute, Solitaire, and Music Quiz
Exact Match: 1
F1 Score: 1.0
---
Question: How many Chinese troops and medics were involved in
the relief efforts?
Answer: 135,000
```

```
Correct Answer: 135,000
Exact Match: 1
F1 Score: 1.0
---
Question: Where did Kanye West first speak about his mother's
death?
Answer: New Zealand
Correct Answer: New Zealand
Exact Match: 1
F1 Score: 1.0
---
Question: When did people first start arriving in the Europea
n continent?
Answer: 45,000 years ago
Correct Answer: 45,000 years ago
Exact Match: 1
F1 Score: 1.0
---
```

Quantitative Evaluation metrics for distilbert-base-cased-distilled-squad model:

For this model the average exact match (EM) score is approximately 0.85, and the median EM score is 1.0. Similarly, the average F1 score is approximately 0.900, and the median F1 score is 1.0. These scores indicate the performance of the model in correctly identifying the answers, which tells us that the model performs very well.



Average EM: 0.85
Median EM: 1.0
Average F1 Score: 0.900574712643678
Median F1 Score: 1.0

Source Code

```
#This code is used from the Homework 10 Lab manual document w
ith my own modifications.

!pip install transformers[torch]

!pip install transformers

!pip install datasets

import pickle

with open('/content/drive/MyDrive/dataset/train_dict.pkl', 'r
b') as f:
    train_dict = pickle.load(f)
with open('/content/drive/MyDrive/dataset/test_dict.pkl', 'r
b') as f:
    test_dict = pickle.load(f)
with open('/content/drive/MyDrive/dataset/eval_dict.pkl', 'r
b') as f:
    eval_dict = pickle.load(f)

with open('/content/drive/MyDrive/dataset/train_data_processe
d.pkl', 'rb') as f:
    train_processed = pickle.load(f)
with open('/content/drive/MyDrive/dataset/test_data_processe
d.pkl', 'rb') as f:
    test_processed = pickle.load(f)
with open('/content/drive/MyDrive/dataset/eval_data_processe
```

```
d.pkl', 'rb') as f:
    eval_processed = pickle.load(f)

print(train_dict.keys())
print(test_dict.keys())
print(eval_dict.keys())

print(train_processed.keys())
print(test_processed.keys())
print(eval_processed.keys())

from transformers import BertForQuestionAnswering

model_name = 'bert-base-uncased'
model = BertForQuestionAnswering.from_pretrained(model_name)

print(model._modules)


from transformers import TrainingArguments

training_args = TrainingArguments(
    output_dir='./results',  # output directory
    num_train_epochs=21,  # total number of training epochs
    per_device_train_batch_size=8,  # batch size per device d
uring training
    per_device_eval_batch_size=8,  # batch size for evaluatio
n
    weight_decay=0.01,  # strength of weight decay
    logging_dir='./logs',  # directory for storing logs
)




from transformers import Trainer
```

```
from datasets import Dataset
import pandas as pd
from transformers import TrainerCallback
from transformers.trainer_utils import IntervalStrategy
import numpy as np



train_dataset = Dataset.from_pandas(pd.DataFrame(train_proces
sed))
eval_dataset = Dataset.from_pandas(pd.DataFrame(eval_processe
d))
test_dataset = Dataset.from_pandas(pd.DataFrame(test_processe
d))


# Custom callback class
class EpochLoggingCallback(TrainerCallback):
    def on_epoch_end(self, args, state, control, **kwargs):
        epoch = state.epoch
        logs = state.log_history[-1]  # Get the latest logged
metrics

        # Extract the loss and learning rate from the Trainer
        loss = logs.get("loss")
        learning_rate = logs.get("learning_rate")

        # Print the loss, learning rate, and epoch informatio
n
        print(f"{{'loss': {loss}, 'learning_rate': {learning_
rate}, 'epoch': {epoch}}}")

# Initialize Trainer
trainer = Trainer(
    model=model,
    args=training_args,
```

```
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
)

# Attach the custom logging callback to the Trainer to print
after each epoch
trainer.add_callback(EpochLoggingCallback())

# Train the model
trainer.train()



import numpy as np

def compute_exact_match(prediction, truth):
    return int(prediction == truth)

def f1_score(prediction, truth):
    pred_tokens = prediction.split()
    truth_tokens = truth.split()

    # if either the prediction or the truth is no-answer then
F1 = 1 if they agree, 0 otherwise
    if len(pred_tokens) == 0 or len(truth_tokens) == 0:
        return int(pred_tokens == truth_tokens)

    common_tokens = set(pred_tokens) & set(truth_tokens)

    # if there are no common tokens then F1 = 0
    if len(common_tokens) == 0:
        return 0

    prec = len(common_tokens) / len(pred_tokens)
    rec = len(common_tokens) / len(truth_tokens)
```

```
    return 2 * (prec * rec) / (prec + rec)


from transformers import BertTokenizer

# Initialize the tokenizer
tokenizer = BertTokenizer.from_pretrained('bert-base-uncase
d')


x = trainer.predict(test_dataset)
start_pos, end_pos = x.predictions
start_pos = np.argmax(start_pos, axis=1)
end_pos = np.argmax(end_pos, axis=1)

# Initialize lists to store EM and F1 scores
em_scores = []
f1_scores = []

# Iterate over predictions
for k, (i, j) in enumerate(zip(start_pos, end_pos)):
    # Get tokens for the current prediction
    tokens = tokenizer.convert_ids_to_tokens(test_processed
['input_ids'][k])

    # Convert token list to string
    predicted_answer = tokenizer.convert_tokens_to_string(tok
ens[i:j+1]).lower()
    correct_answer = test_dict['answers'][k]['text'][0].lower
()

    # Compute EM and F1 scores
    em = compute_exact_match(predicted_answer, correct_answe
r)
    f1 = f1_score(predicted_answer, correct_answer)
```

```
    # Append scores to lists
    em_scores.append(em)
    f1_scores.append(f1)

    # Print results for individual prediction
    print('Question:', test_dict['question'][k])
    print('Answer:', predicted_answer)
    print('Correct Answer:', correct_answer)
    print('Exact Match:', em)
    print('F1 Score:', f1)
    print('---')

# Calculate average and median scores
avg_em = np.mean(em_scores)
median_em = np.median(em_scores)
avg_f1 = np.mean(f1_scores)
median_f1 = np.median(f1_scores)

# Print average and median scores
print('Average EM:', avg_em)
print('Median EM:', median_em)
print('Average F1 Score:', avg_f1)
print('Median F1 Score:', median_f1)



from transformers import pipeline

# Initialize question answering pipeline
question_answerer = pipeline("question-answering", model='dis
tilbert-base-cased-distilled-squad')

# Initialize lists to store EM and F1 scores
em_scores1 = []
f1_scores1 = []
```

```
# Iterate over test questions
for i in range(len(test_dict['question'][:20])):
    # Get prediction from question answering model
    result = question_answerer(question=test_dict['question']
[i], context=test_dict['context'][i])

    # Compute EM and F1 scores for the prediction
    em = compute_exact_match(result['answer'], test_dict['ans
wers'][i]['text'][0])
    f1 = f1_score(result['answer'], test_dict['answers'][i]
['text'][0])

    # Append scores to lists
    em_scores1.append(em)
    f1_scores1.append(f1)

    # Print results for individual question
    print('Question:', test_dict['question'][i])
    print('Answer:', result['answer'])
    print('Correct Answer:', test_dict['answers'][i]['text']
[0])
    print('Exact Match:', em)
    print('F1 Score:', f1)
    print('---')

# Calculate average and median scores
avg_em = np.mean(em_scores1)
median_em = np.median(em_scores1)
avg_f1 = np.mean(f1_scores1)
median_f1 = np.median(f1_scores1)

# Print average and median scores
print('Average EM:', avg_em)
print('Median EM:', median_em)
```

```
print('Average F1 Score:', avg_f1)
print('Median F1 Score:', median_f1)
```