Department of Computer Science and Engineering (Data Science) Lab Manual

Sub: Advanced Computational Linguistics

Experiment No 9

NAME:DIVYESH KHUNT

SAPID:60009210116

BATCH:D12

Year/Sem: BTech/VII

Aim: To fine-tune a BERT model on custom data for performing a question-answering task.

Introduction:

Question Answering (QA) systems aim to provide precise answers to questions posed in natural language. Fine-tuning BERT (Bidirectional Encoder Representations from Transformers) on custom datasets allows for developing QA models tailored to specific domains. By training on relevant question-answer pairs, the model learns to locate and provide precise answers based on context.

BERT is especially effective for this task due to its ability to understand contextual embedding's, making it well-suited for handling the complexities of language. Fine-tuning involves training the model on a labeled dataset where each entry includes a passage, a question, and the corresponding answer span within the passage.

Fine-Tuning BERT for Question Answering

- 1. **Model Selection**: Use the pre-trained BERT model as the base, available through Hugging Face's Transformers library. Models such as bert-large-uncased are commonly used for QA tasks.
- 2. **Data Preparation**: Format the custom data into question-passage-answer triplets. The passage contains the context, and the answer is a span within this context.
- 3. **Tokenization**: Tokenize the data using the BERT tokenizer, which splits text into tokens compatible with BERT's vocabulary. Additionally, align each token with its position in the original text to aid in identifying the answer span.
- 4. **Fine-Tuning Process**: Train the BERT model on the formatted dataset, optimizing it to locate the start and end tokens of the answer span within the context passage.

Lab Experiment

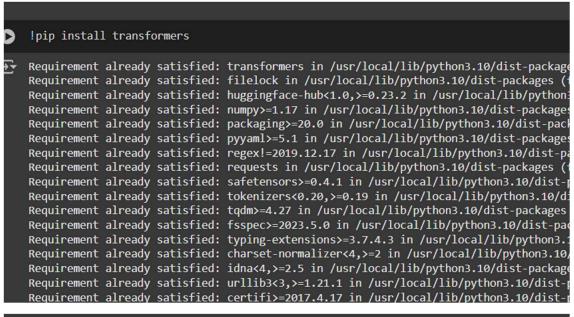
- **Step 1**: Install required libraries.
- **Step 2**: Load a pre-trained BERT model for QA and tokenizer.
- **Step 3**: Prepare the custom data in question-passage-answer format, ensuring each answer is marked with its start and end positions in the passage.
- **Step 4**: Tokenize the data and format it for BERT.
- **Step 5**: Fine-tune the model.

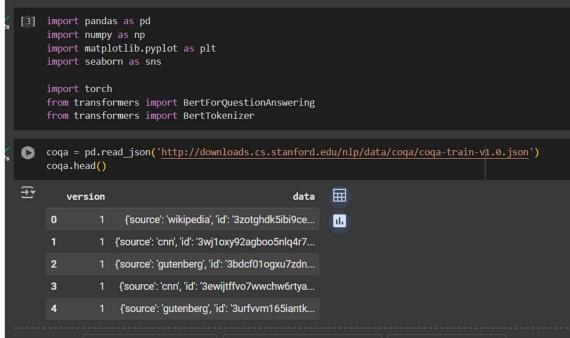


Year/Sem: BTech/VII

Department of Computer Science and Engineering (Data Science) Lab Manual

Step 6: Test the model with new questions and passages.



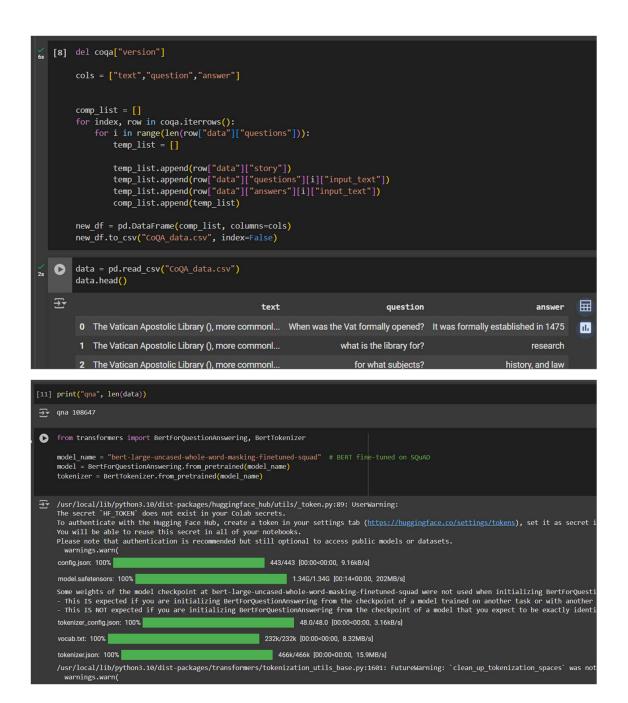


(Autonomous College Affiliated to the University of Mumbai) NAAC Accredited with "A" Grade (CGPA: 3.18)



Year/Sem: BTech/VII

Department of Computer Science and Engineering (Data Science) Lab Manual



(Autonomous College Affiliated to the University of Mumbai) NAAC Accredited with "A" Grade (CGPA: 3.18)



Year/Sem: BTech/VII

Department of Computer Science and Engineering (Data Science) Lab Manual

```
[15] model = BertForQuestionAnswering.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')
tokenizer = BertTokenizer.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')

Some weights of the model checkpoint at bert-large-uncased-whole-word-masking-finetuned-squad were not used when initializing BertFor_This IS expected if you are initializing BertForQuestionAnswering from the checkpoint of a model trained on another task or with a This IS NOT expected if you are initializing BertForQuestionAnswering from the checkpoint of a model that you expect to be exactly question = data["question"][random_num]
text = data["question"][random_num]
answer = data["answer"][random_num]
print(text)
print(question)
print(question)
print(answer)

Shane Thomas is a 10-year-old pianist from England. He's being called the next Mozart because of his amazing abilities.
He has only been having piano lessons for four months, and practices four hours a week, but he has already played difficult classical when Shane was three years old, he said that he could play the piano, but nobody took him seriously. At school, he could listen to the tis tutor, Richard Goffin-Lecar, says he is like Amadeus Mozart, who lived during the 18thcentury in Salzburg, Austria, and was one His father, a single parent with two other children, says that although he has little money, he wants to send Shane to a good music When did his lessons start?

four months ago
```



(Autonomous College Affiliated to the University of Mumbai) NAAC Accredited with "A" Grade (CGPA: 3.18)



Year/Sem: BTech/VII

Department of Computer Science and Engineering (Data Science) Lab Manual

```
tokens = tokenizer.convert_ids_to_tokens(input_ids)
for token, id in zip(tokens, input_ids):
    print('{:8}{:8,}'.format(token,id))
           2,769
money
           1,010
he
           2,002
wants
           4,122
to
           2,000
           4,604
send
shane
           8,683
to
           2,000
           1,037
           2,204
good
music
           2,189
school
           2,082
           1,012
           1,000
           1,045
           1,005
m
           1,049
           1,037
single
           2,309
father
           2,269
           1,010
but
           2,021
```

(Autonomous College Affiliated to the University of Mumbai) NAAC Accredited with "A" Grade (CGPA: 3.18)



Year/Sem: BTech/VII

Department of Computer Science and Engineering (Data Science) Lab Manual

```
[29] output = model(torch.tensor([input_ids]), token_type_ids=torch.tensor([segment_ids]))

answer_start = torch.argmax(output.start_logits)
answer_end = torch.argmax(output.end_logits)
if answer_end >= answer_start:
    answer = " ".join(tokens[answer_start:answer_end+1])
else:
    print("I am unable to find the answer to this question. Can you please ask another question?")

print("\nQuestion:\n{}".format(question.capitalize()))

print("\nAnswer:\n{}.".format(answer.capitalize()))

Question:
    When did his lessons start?

Answer:
Four months.
```

```
def question_answer(question, text):
    input ids = tokenizer.encode(question, text)
    tokens = tokenizer.convert_ids_to_tokens(input_ids)
    #first occurence of [SEP] token
    sep idx = input_ids.index(tokenizer.sep_token_id)
    num seg a = sep idx+1
    num seg b = len(input ids) - num seg a
    segment ids = [0]*num seg a + [1]*num seg b
    assert len(segment_ids) == len(input_ids)
    #model output using input_ids and segment_ids
    output = model(torch.tensor([input_ids]), token_type_ids=torch.tensor([segment_ids]))
    answer_start = torch.argmax(output.start_logits)
    answer_end = torch.argmax(output.end_logits)
    if answer_end >= answer_start:
        answer = tokens[answer_start]
        for i in range(answer_start+1, answer_end+1):
            if tokens[i][0:2] == "##":
                answer += tokens[i][2:]
                answer += " " + tokens[i]
    if answer.startswith("[CLS]"):
        answer = "Unable to find the answer to your question."
    print("\nPredicted answer:\n{}".format(answer.capitalize()))
```

(Autonomous College Affiliated to the University of Mumbai) NAAC Accredited with "A" Grade (CGPA: 3.18)



Year/Sem: BTech/VII

Department of Computer Science and Engineering (Data Science) Lab Manual

```
[35] text = """New York (CNN) -- More than 80 Michael Jackson collectibles -- including the late pop star's famous rhinestone-studded glove f question = "Where was the Auction held?"
question_answer(question, text)
#original answer from the dataset
print("Original answer:\n", data.loc[data["question"] == question]["answer"].values[0])

**Predicted answer:
Hard rock cafe in new york 's times square
Original answer:
Hard Rock Cafe
```

```
text = input("Please enter your text: \n")
    question = input("\nPlease enter your question: \n")
        question_answer(question, text)
        flag = True
        flag_N = False
        while flag:
            response = input("\nDo you want to ask another question based on this text (Y/N)? ")
            if response[0] == "Y":
                question = input("\nPlease enter your question: \n")
                flag = False
            elif response[0] == "N":
                print("\nBye!")
                flag = False
                flag_N = True
        if flag_N == True:
••• Please enter your text:
    The Vatican Apostolic Library, often referred to simply as the Vatican Library, is the library of the Holy Se
    Please enter your question:
    When was the Vatican Apostolic Library founded?
    Predicted answer:
    15th century
    Do you want to ask another question based on this text (Y/N)?
```