**Comp5211 2019-Fall Final Project**

**COMP5211**

**WANG, Sukai 20528424**

Problem 3 - NLP:

Evaluating pretrained language model: BERT

**Introduction**

BERT[1] (Bidirectional Encoder Representations from Transformers) is a state-of-the-art method in the area of NLP task.

Transformer is used in BERT as an attention mechanism to learn the contextual relations between words and sentences. The transformer can be split into two parts, one is encoder and the other is decoder. Encoder can consume text input and decoder can output the predict result in different tasks. It can be considered as a bidirectional text model since it reads the entire sequence at once, rather than read sequence from left to right or from right to left.

**Experiment Results:**

1. Environment:

I use docker to create a valid environment. Docker is the

Docker is a popular tool to make it easier to build, deploy and run applications using containers

I placed my dockerfile in docker/dockerfile

The docker image and container created are shown below:

Docker image:



Docker container:



1. Bert server:

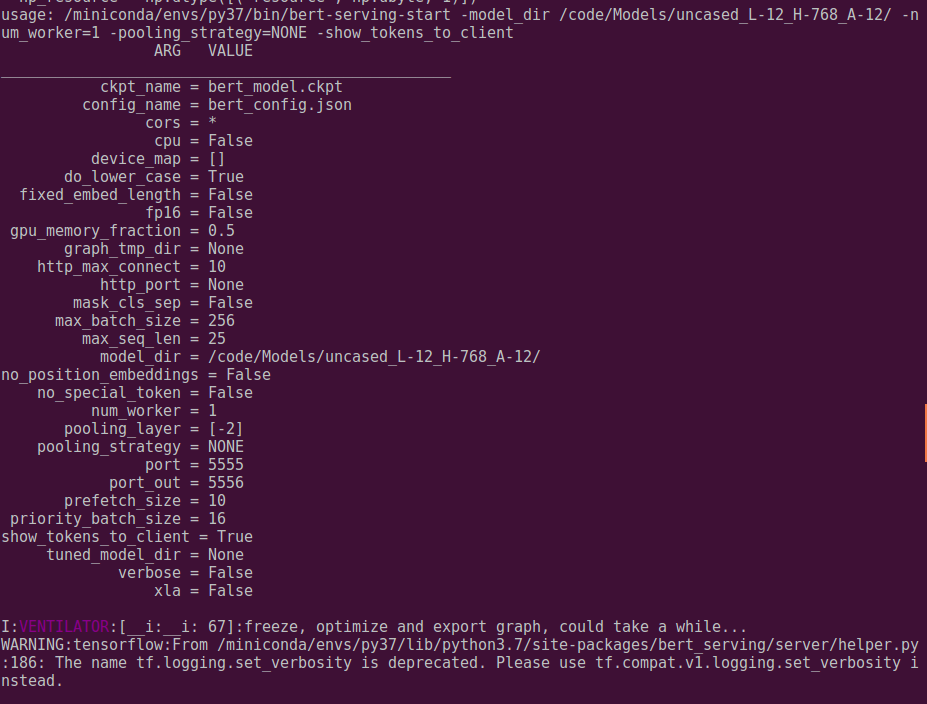
I downloaded the model from <https://github.com/CodePothunter/COMP5211_Project/blob/master/Bert-as-Service_Install.md> and placed it in Model folder.

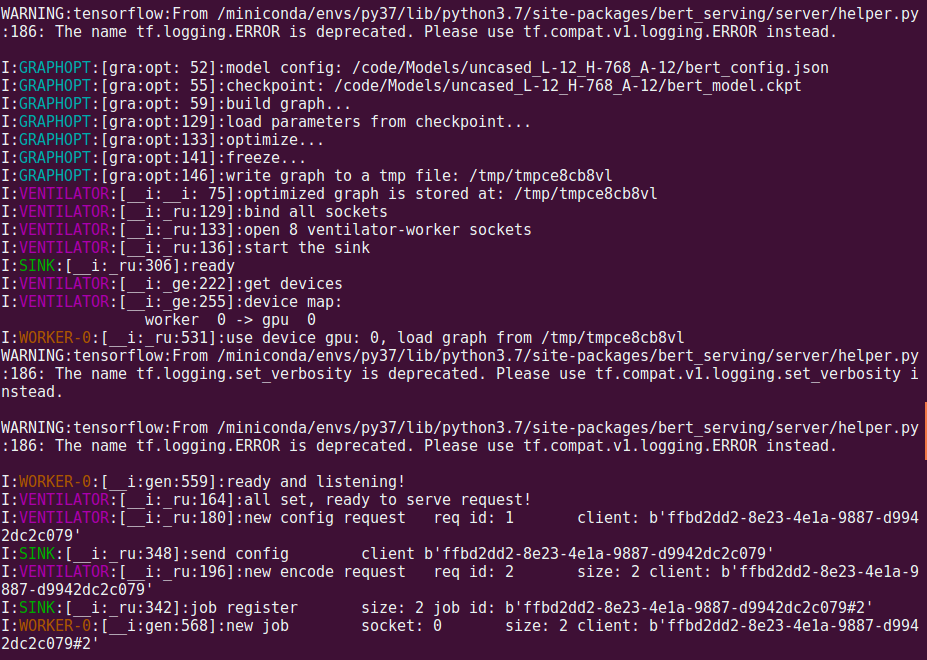
Then I use

*bert-serving-start -model\_dir /code/Models/uncased\_L-12\_H-768\_A-12/ -num\_worker=1 -pooling\_strategy=NONE -show\_tokens\_to\_client*

to start the server.

The starting and responding interfaces are shown in below.





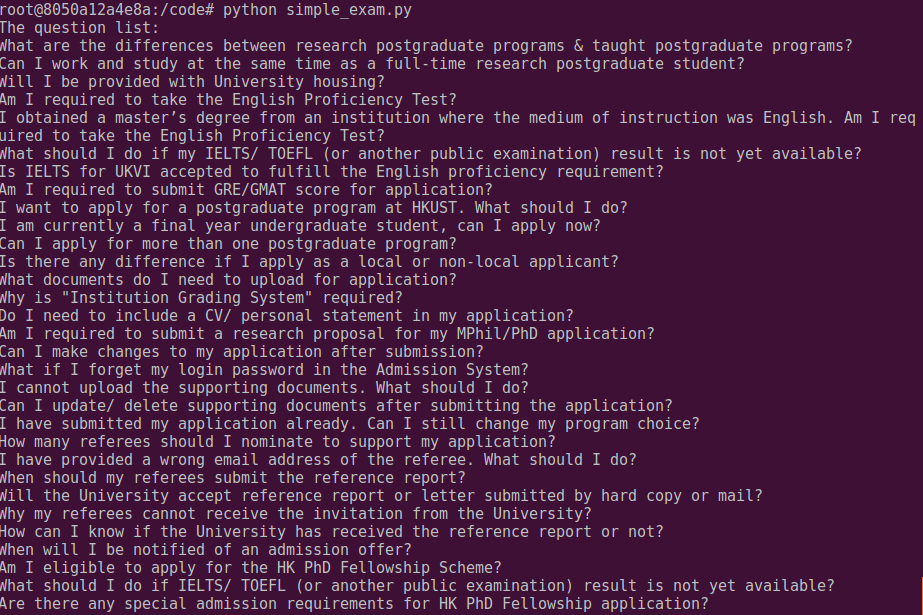
1. Q&A

I used example.txt as the evaluation content. The content contains lots of lines, where each line use ||| to split the question and answer.

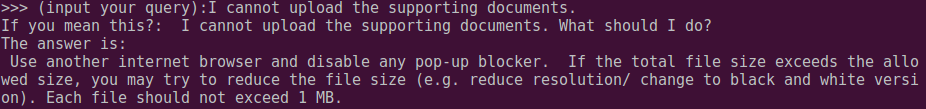
Our code use BERT to classify each question(encode each question language), and when we input the query sentence, we use BERT to encode query sentence too. Then we can find the closest question sentence, which means this question has the closest meaning with the query problem.

The results are shown below:

Firstly, I output the entire question list.

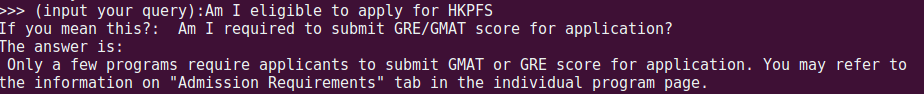


Every query sentence will be found a closest meaning question and output the prepared answer as the query’s answer.

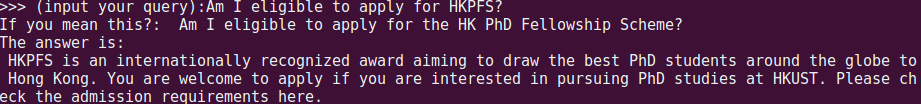


We can see that though the query is not as same as the prepared question, we can get the correct answer successfully. That’s mean the BERT has the ability to find the sentences’ meaning.

But we can also find some interesting results:



When I asked “Am I eligible to apply for HKPFS”, I got the closest problem “Am I required to submit GRE/GMAT score for application?”. But when I input the query with “?” at the end of it, then we can get what we want.



1. More Evaluations – NOT test:

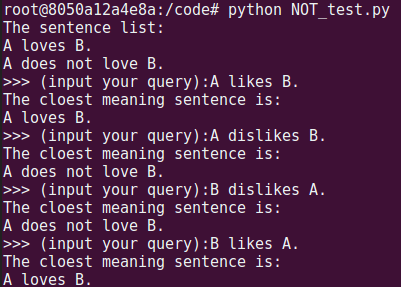
In this part, we evaluate the BERT model’s ability in distinguishing the “NOT” in a sentence.

When we set the sentence list:

*A loves B.*

*A does not love B.*

We can test the BERT results:



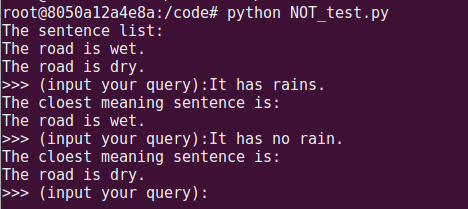
Then we set the sentence list as:

*The road is wet.*

*The road is dry.*

So, when we input that “It has rains.” BERT will understand that the road is wet.

And when we input the “NOT”, BERT will think the road is dry.



Lastly we test the “Apple” question.

We set the sentence list as:

*We should eat apple.*

*We should not eat apple.*

And when we input that: “Apple is good for our health.”

It will output that: We should eat apple.

And when we input that: ”Apple is bad for our health.”

It will connect the mean of “bad” with “not”, and output “We should not eat apple.”

