Question Answering

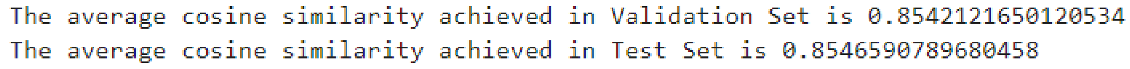
Submitted by – Sharanya Bhattacharya

Task Understanding - To my understanding, this task is asking for Semantic Search over the Question space. This makes sense since, the dataset doesn't contain a context field which is typical in Extractive or Abstractive Question Answering datasets such as SQuAD. Also, since Medical QnA has high impact risk, we should not try to use GenAI carelessly, as it may hallucinate and lead to spurious results. Even if we want to use LLMs, we should place strong guardrails in place for the same. Thus, I have made my approaches based on this assumption of Semantic Search.

Dataset Understanding – The given dataset contains 16407 Questions and Answers in the Medical Domain. The important fact to note is that it does not contain a context field from which the answer is generated. To this end, the dataset is not really typical for QnA. The dataset additionally contains, a qtype field which I have not used for the task.
Additional EDA has been performed and the results have been given the attached Notebook file.
After data cleaning, I have divided the dataset into train, validation and test sets in the ratio of 70:15:15.

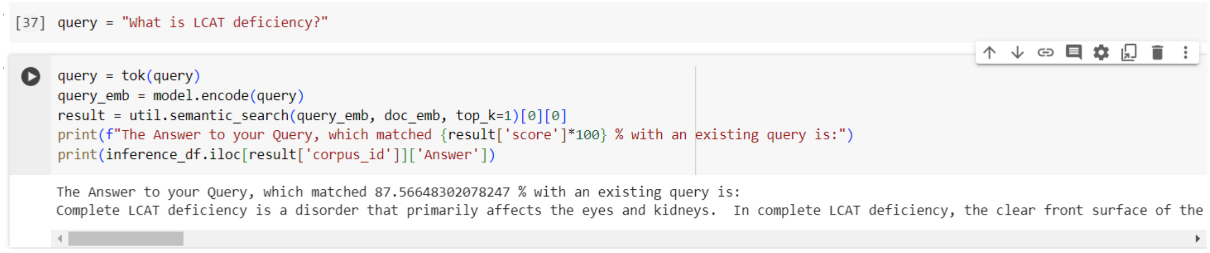
Approach Used – To tackle this problem of Semantic Search, I was initially trying out a BiLSTM based representation search, but then decided against it on further research into Semantic Search.
Today, the state of art uses the Sentence Transformer approach, which converts an entire sentence into some specified dimensional vector representation. The vector representations are also generally dense. For my current submission I have used the *multi-qa-MiniLM-L6-cos-v1* pretrained sentence transformer, which is specially trained for semantic search. It returns a 384-dimensional representation for each sentence.
What I have done, is cleaned each Question in the training sent by myself, and then found their representations using the aforementioned model. Then for a Inference, I have cleaned the Question using the same rules, and found its representation. Then as the total number of documents is quite low (~11k in the training dataset) we can easily search for the closest *Cosine Similarity Match* and quite quickly at that. I have done exactly that and returned the answer of the most similar question available to us, as the answer for the query given to us.

Evaluation – The evaluation of this assignment was the most challenging for me. In the given short amount of time I could not decide on the best method of evaluating the Semantic Search approach used. I have however found out the average cosine similarity of queries to the documents whose answers I have returned in the Validation and Test sets. According to few articles I read online, we can approximately consider two documents to be quite close if we have a cosine similarity of at least 0.7. I am happy to report that for my QnA set average cosine similarity is greater than 0.8. As shown below(and in the Notebook)



```
The average cosine similarity achieved in Validation Set is 0.8542121650120534
The average cosine similarity achieved in Test Set is 0.8546590789680458
```

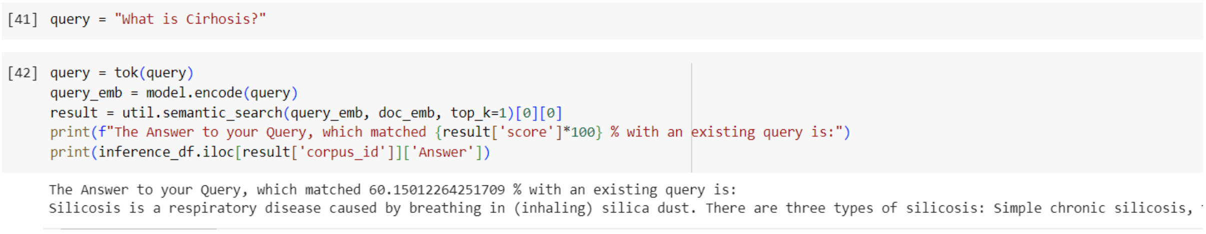
Examples of General Inferences:



```
[37] query = "What is LCAT deficiency?"
```

```
query = tok(query)
query_emb = model.encode(query)
result = util.semantic_search(query_emb, doc_emb, top_k=1)[0][0]
print(f"The Answer to your Query, which matched {result['score']*100} % with an existing query is:")
print(inference_df.iloc[result['corpus_id']]['Answer'])
```

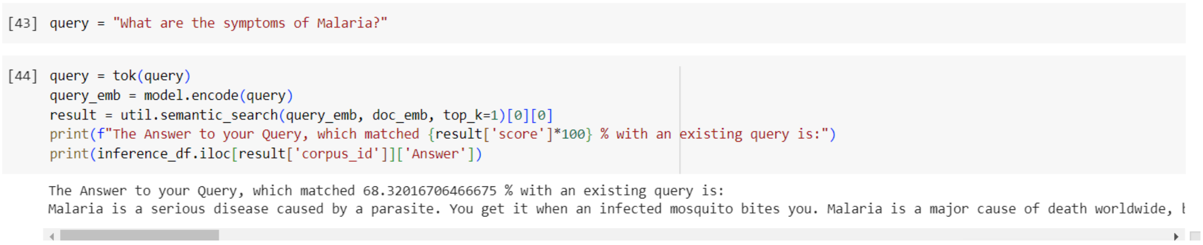
```
The Answer to your Query, which matched 87.56648302078247 % with an existing query is:
Complete LCAT deficiency is a disorder that primarily affects the eyes and kidneys.  In complete LCAT deficiency, the clear front surface of the
```



```
[41] query = "What is Cirhosis?"
```

```
[42] query = tok(query)
     query_emb = model.encode(query)
     result = util.semantic_search(query_emb, doc_emb, top_k=1)[0][0]
     print(f"The Answer to your Query, which matched {result['score']*100} % with an existing query is:")
     print(inference_df.iloc[result['corpus_id']]['Answer'])
```

```
The Answer to your Query, which matched 60.15012264251709 % with an existing query is:
Silicosis is a respiratory disease caused by breathing in (inhaling) silica dust. There are three types of silicosis: Simple chronic silicosis,
```



```
[43] query = "What are the symptoms of Malaria?"
```

```
[44] query = tok(query)
     query_emb = model.encode(query)
     result = util.semantic_search(query_emb, doc_emb, top_k=1)[0][0]
     print(f"The Answer to your Query, which matched {result['score']*100} % with an existing query is:")
     print(inference_df.iloc[result['corpus_id']]['Answer'])
```

```
The Answer to your Query, which matched 68.32016706466675 % with an existing query is:
Malaria is a serious disease caused by a parasite. You get it when an infected mosquito bites you. Malaria is a major cause of death worldwide,
```

What could be improved and Future Work – In my opinion the results are good as a first approach. However, we can improve upon the pipeline in a few ways:

1. We must try a few different other sentence transformer models to see which one works best for our dataset.
2. Instead of directly returning the best match, we can use some top k matches and then use them as context to generate answers using RAG or finetuned LLMs
3. Evaluation needs to be done more rigorously and using better approaches which I could not find.