Brance Research InternTask

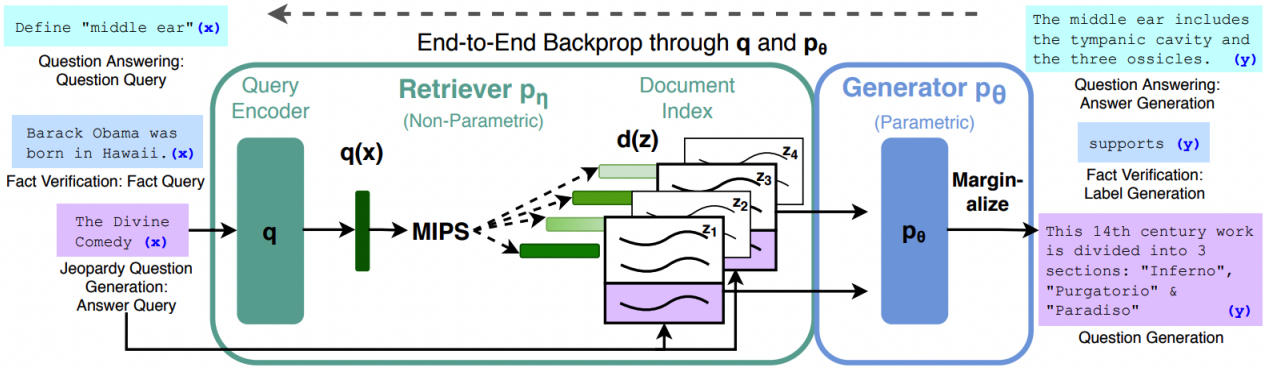
Name:Harshul Surana
Linkedin Profile: linkedin
Date Challenge Received:06-07-23
Date Solution Delivered:14-07-23

1. Problem Statement

The problem statement for the first approach is to use the RAG model, introduced by Lewis et al in 2019, to answer questions based upon the given knowledge document (information about PAN cards). The problem statement is closed domain Retrieval-based Question Answering and the goal is to maximise the “accuracy” between the model output and the desired output (gold standard or eval data). The implementation is based on the above paper.

2. Approach

My approach to this problem was to use HuggingFace library to instantiate the pre-trained model and run inference.
The implicit assumption is that we are NOT fine-tuning the RAG model. This is because we do not have ample training data - the knowledge document provided is not in the form of question-answer pairs. We can convert the document to this form, but the number of pairs will be limited and not enough to fine-tune the model (generally atleast ~1000 data points needed for fine-tuning BERT) and will require human intervention to form the answers. Therefore this approach is about running the model on inference to get accurate answers.



3. Solution

a) Dense Passage Retrieval (DPR)

The document is read and chunked into 100 token long chunks. The total number of chunks are 42. The chunks are then tokenized using "facebook/dpr-ctx_encoder-multiset-base" which is the context encoder trained using the Natural Questions (NQ) dataset, TriviaQA, WebQuestions (WQ), and CuratedTREC (TREC).

The next step is to create an index for these 42 documents. This is done using FAISS. Faiss is a library for efficient similarity search and clustering of dense vectors. It contains algorithms that search in sets of vectors of any size.

The questions are tokenized and encoded using dpr-question_encoder-multiset-base, which is the question encoder trained using the Natural Questions (NQ) dataset, TriviaQA, WebQuestions (WQ), and CuratedTREC (TREC).

DPR finds the k most similar documents to the given question, using MIPS.

k is a hyperparameter and set to 3. This means for a given question, the 3 most similar documents are selected.

b) Retriever

The Retriever is used to get documents from vector queries. It retrieves the documents embeddings as well as the documents contents, and it formats them to be used with a RagModel.

There are two variants of RAG: RAG-Sequence and RAG-Token. The rag-sequence-nq model is used here because it seems to give better answers.

RAG-Sequence is less compute intensive, but it can't combine information from multiple passages. Instead, for each of the top 'k' passages returned by the Retriever, the Generator produces a separate answer, and then RAG chooses the most probable answer among the 'k' options.

c) Generator

The top-k contextualized inputs is passed to the generator. BART is the generator used. The maximum length and minimum token length is set to **50 and 5** respectively.

4. Results and Evaluation

```
Q: What is the cost/fees of a PAN card?
A: ' us $ 299.90'

Response took 183.50 seconds
2
Q: Can I take the delivery of Pan card at Indian address
A: ' only at an indian address'

Response took 187.19 seconds
3
Q: How long does it usually take to receive the PAN card after applying?
A: ' around 2 - 3 weeks'

Response took 224.00 seconds
4
Q: How to apply for PAN card
A: ' through abc'
```

As shown, the generated output is not very desirable. The model chooses to answer the questions using 2-4 words. In the best case, the output is not nuanced as it does not give the user the different scenarios, and for some questions the answer is gibberish.

I plan to use **Cosine Similarity** as the evaluation metric, using the SBERT.

The idea is that the more accurate generated answer will be more semantically similar to to the gold answer(y), and therefore have a higher cosine similarity score. The generated and the gold answer are passed to SBERT to get the respective sentence embeddings, and the cosine similarity is calculated.

While traditional evaluation metrics like exact match is used (SQuAD), it will not be useful in this case because the answers are generative in nature. Instead of penalizing the model for not outputting the exact words, the semantic similarity to the gold answer should give a good indication of the accuracy of the generated answer.

4. Future Scope

As shown, the traditional RAG model is not good for this task. Future approaches include experimenting with the latest SOTA generative models to generate high quality answers, and to enable the user to chat with the system. These approaches will be taken up in the subsequent write-up