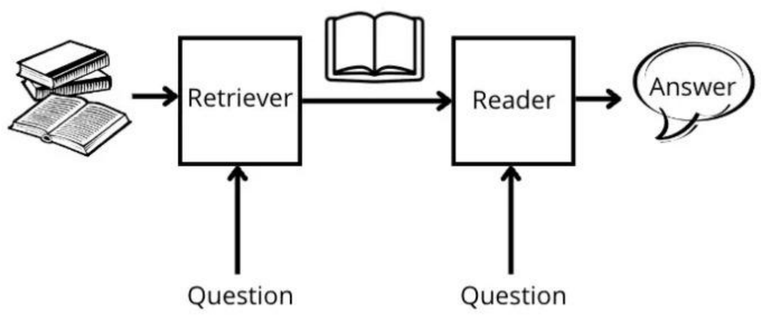
Dense Passage Retrieval

Qiming Wu, Zihan Ma, Shi Bu

1 Problem Definition

Open Domain Question Answering system reply on the efficient passage retrieval methods. This step helps selecting the relevant contexts for answering questions. This system follows two steps:

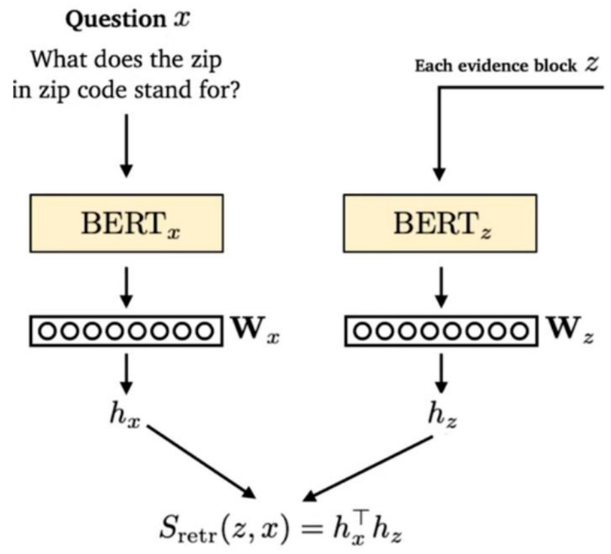
- **Context Retriever**: give certain number of relevant passages (focus in this report)
- **Machine Reader**: Identify the correct answer from the passages



QA system using TF-IDF or BM25 to match key words efficiently with an inverted index. However, TF-IDF and BM25 have difficulty retrieving context and identifying synonyms. So we propose to use learned dense embeddings so that synonyms or paraphrases that consist of completely different tokens may still be mapped to vectors close to each other.

2 DPR

Dense Passage Retriever (DPR) is used to fetching the relevant passages w.r.t the question asked based on the similarity between the high-quality low-dimensional representation of passages and questions.



The model uses a dual-encoder architecture (Figure)
x and z are question text and passage text.
h_x and h_z are BERT model that outputs question and passage representation.
The model find the most relevant result by calculating the similarity between the question and passage embeddings by the dot product of them.

3 FAISS

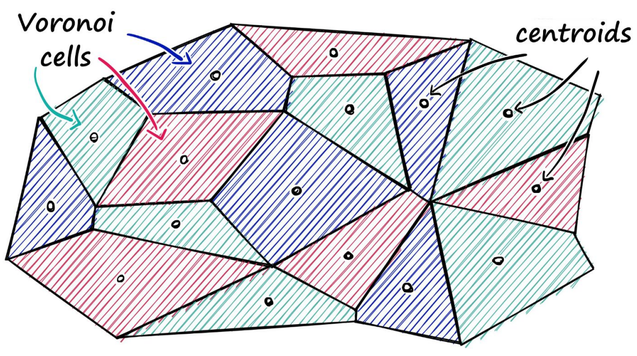
Problem in online settings:

- Performing exact match over millions of vectors is time consuming
- During inference time, they apply the passage encoder Ep to all the passages and index them using FAISS offline.

How FAISS reduce search time:

- Partitioning the index (approximate by reducing the scope of our search)
- Quantization (approximates the distance/similarity calculation)

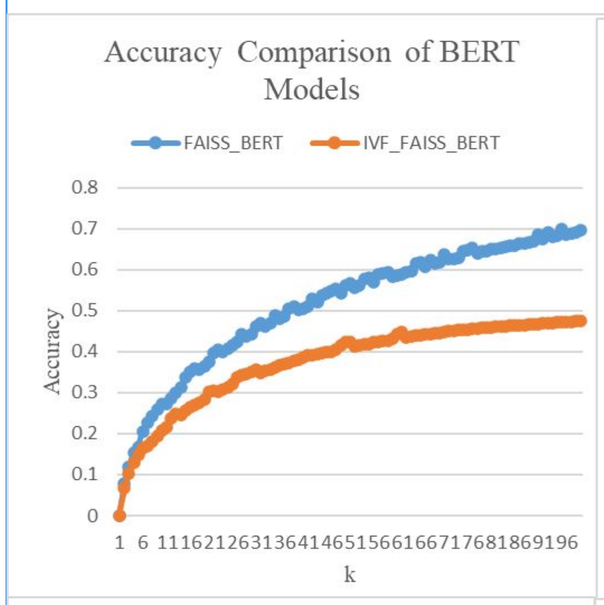
Partition:

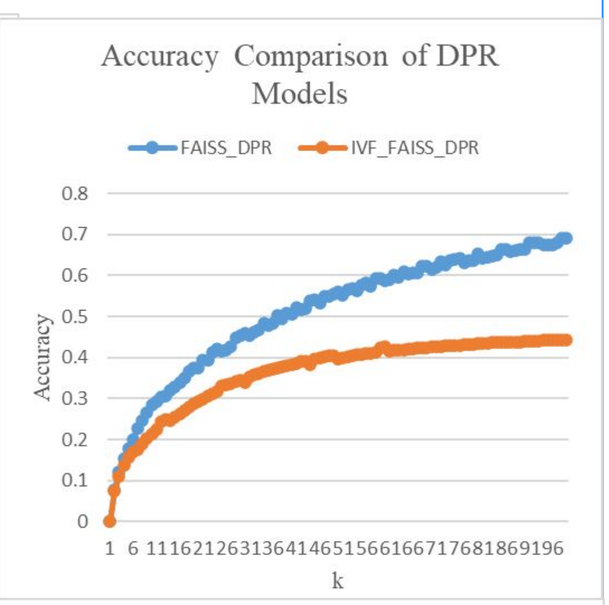


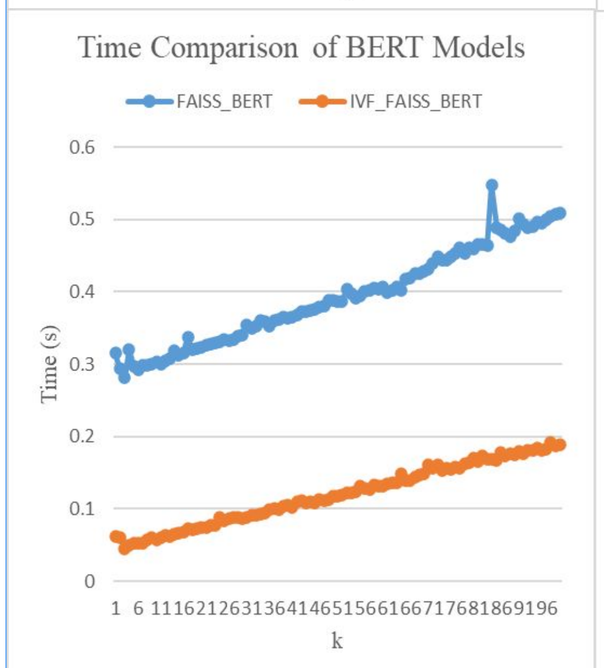
- Partition the index into cells.
- Measure query vector distance between centroids, then restrict our search scope to that centroid's cell.

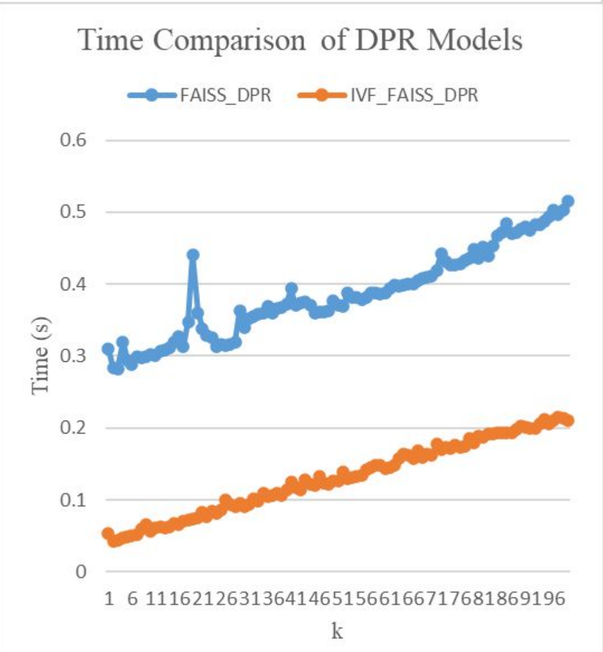
4 DPR Experiment

We first evaluate the BERT and DPR model with FAISS framework, and then we calculate the accuracy of model predictions along with time cost. Here is the summary of the experiments.









This is the first time for DPR and BERT model to inference on QANTA dataset. From the 2 upper figures, we can clearly see that both BERT and DPR models can perform well on the QANTA dataset. Apart from that, we observe that when using the FAISS framework without IVF, the performances of DPR and BERT models are much better than those with IVF (especially when k is large). Also, we compare the inference time of DPR and BERT models with or without IVF. We observe that IVF could accelerate the whole process while giving relatively worse performances when using large k values.

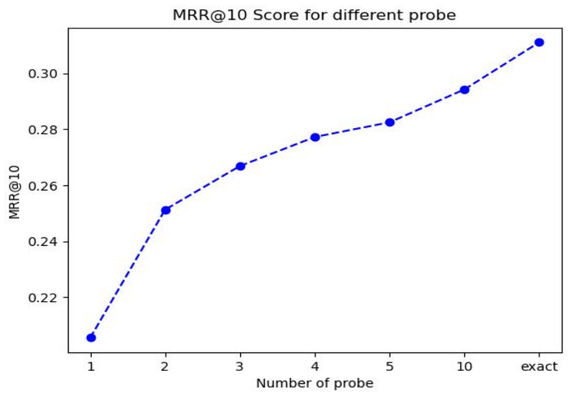
5 FAISS Experiment

Goal: we want to find out is how FAISS partition affect the performance.
Result: we found the FAISS partition have some effect on the rank, but the main performance reduction come from partition.

We use the settings of splitting the embeddings into 256 centroid, and testing with 1 probe case (which meanings restrict search scope to the nearest centroid's cell) on MSMARCO dataset.

The MRR@10 score is 0.311 for exact match and 0.206 using our settings

- In this settings, we found there are only 0.07% cases retrieved by FAISS partition have lower rank comparing to exact match
- However, 43.9% of cases that the true answer are in different centroid with FAISS matched nearest centroid, and, for the questions that retrieved by exact match and not by FAISS partition, 98.1% of cases that the true answer are in different centroid.



6 Future works

In the future, we want to find out is there better way to train dense sentence encoder, so that it can generate better embeddings in approximate Nearest-Neighbor search settings