Inter IIT Tech Meet 11.0

Expert Answers in a Flash: Improving Domain-Specific QA

TEAM ID: 44

Problem statement

This problem is based on extractive question answering. It involves the task of retrieving the paragraphs that can be used to answer a given question and then finding an answer span within the found context paragraph for a given question. Our task was to perform the same on the given dataset and under computational constraints.

Task 1

Paragraph retrieval

Data structure

We segregated paragraphs based on their theme and stored the theme as key-value pair (the theme is the key) for accessing the data related to that theme while searching for paragraphs related to the question.

Embedding Models

We tried multiple models to get the best speed and accuracy tradeoff among multiple text-to-embedding models for the sentence similarity task.

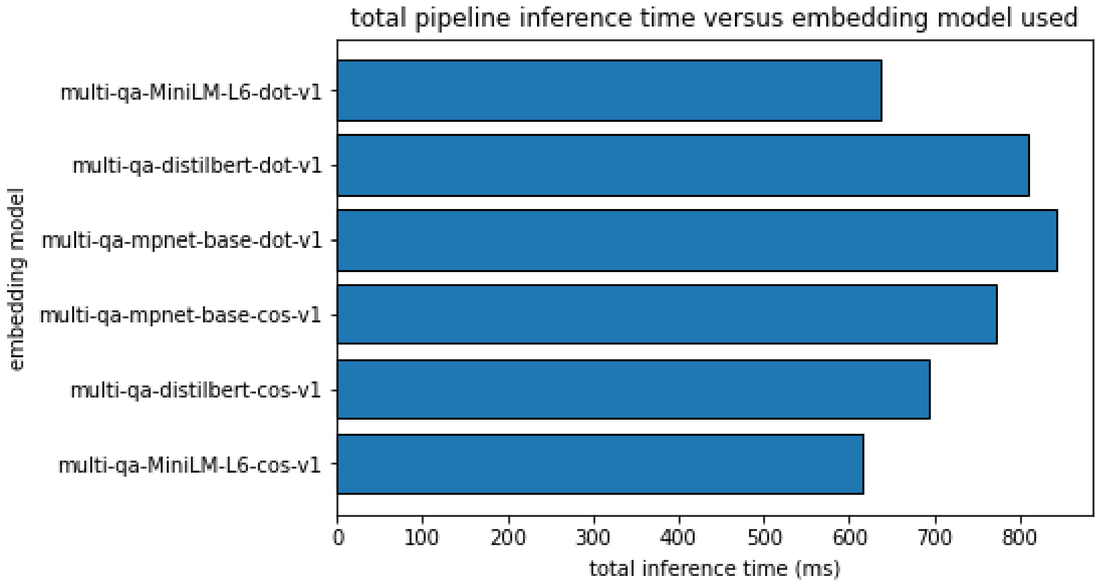
Nearest Neighbour Search

We used the FAISS library to get the approximate Neighbour with some loss in accuracy but a much lower search time.

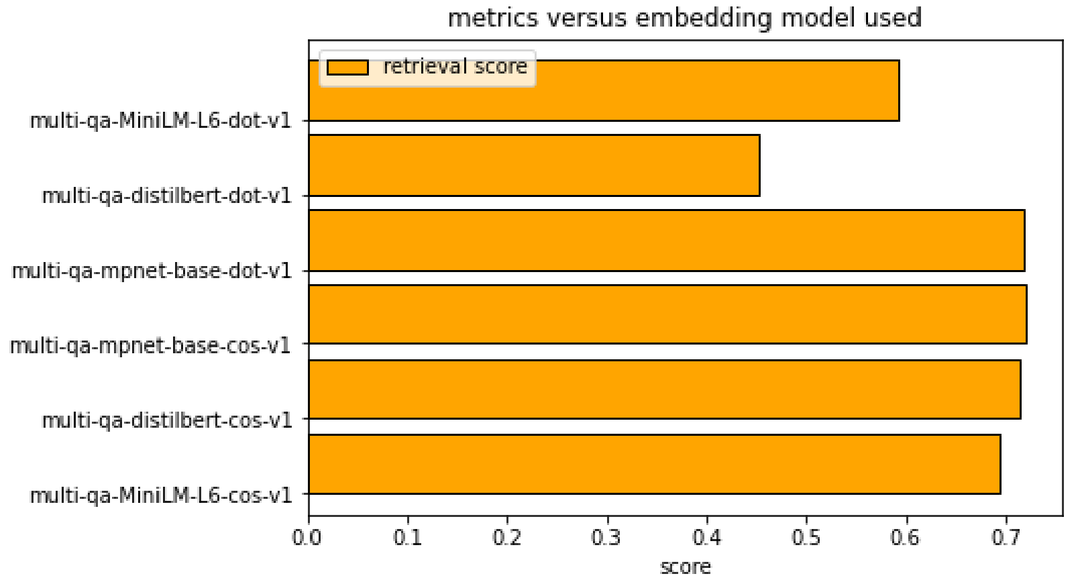
Data Structure

- Selected dictionary as the data structure to store all the data related to a theme(paragraph data frame, FAISS index)
- Dictionaries provide constant time access to the values given the keys
- Used index-based search to get the paragraph from the data frame in constant time.

Embedding Models



Latency

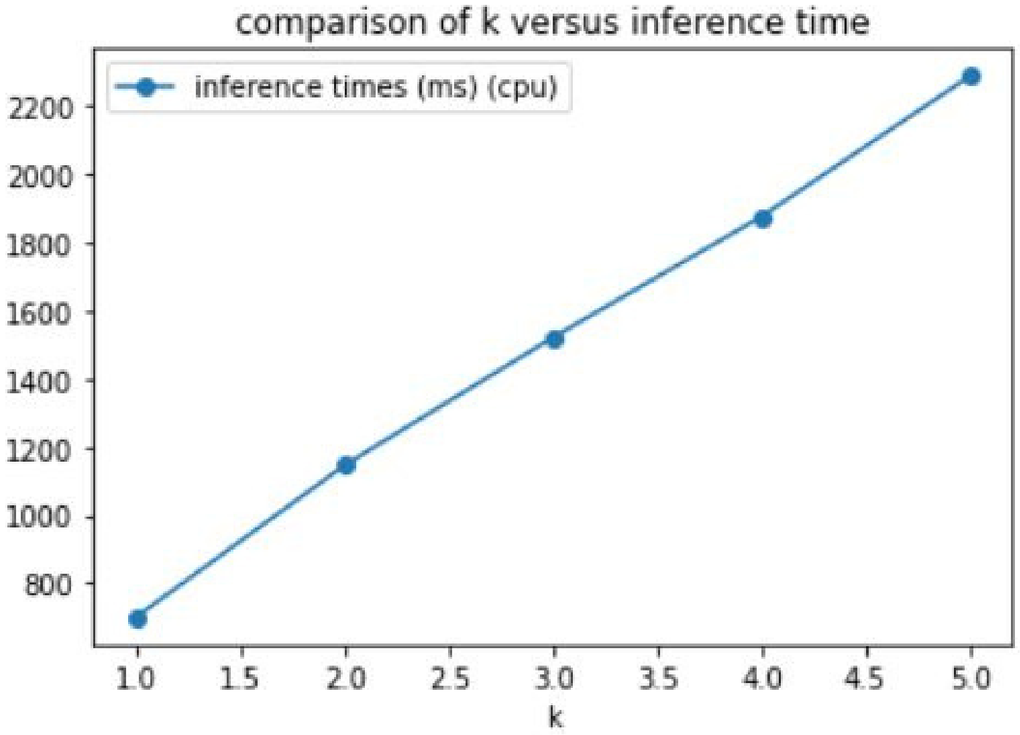


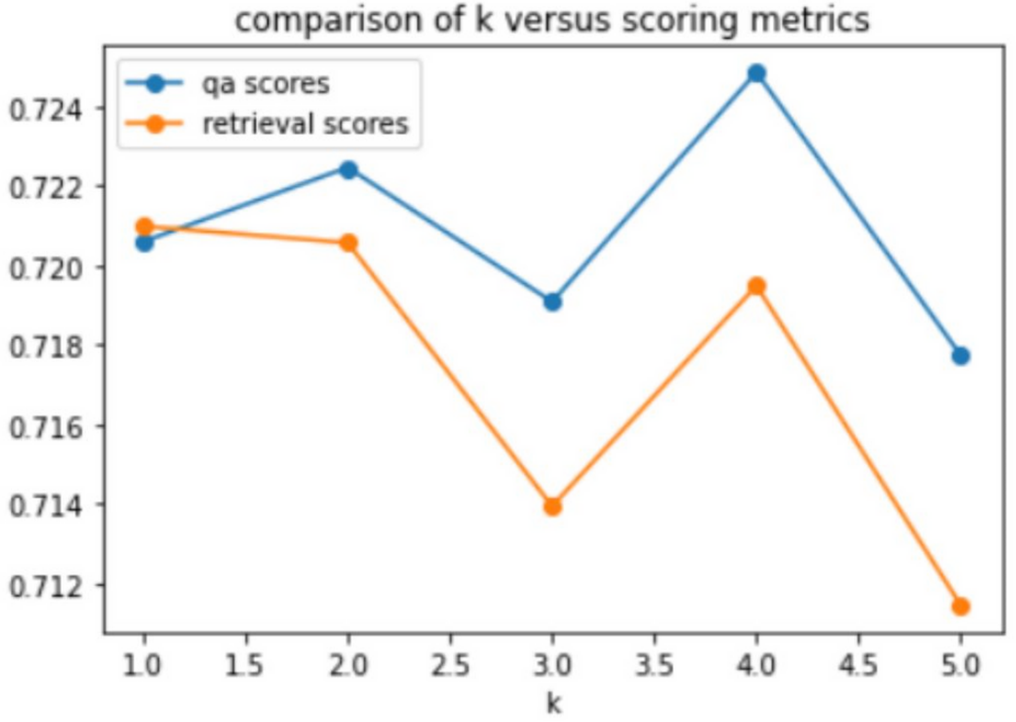
Accuracy Scores

Model Selection

We selected the model multi-qa-mpnet-base-cos-v1 due to its highest accuracy and the inference time which was not an issue as we only scanned one neighbor.

Number of neighbors vs QA Score





Choice of number of neighbours

We decided to look for answers in only 1st paragraph we retrieve as scanning other paragraphs gave marginal improvement along with higher inference time. This would lead to a lower overall score.

Task 2
Question Answering

GPT-based Approach

Used a combination of a pretrained GPT model and Multi-layered Perceptrons to find out the answer probability and the answer locations.

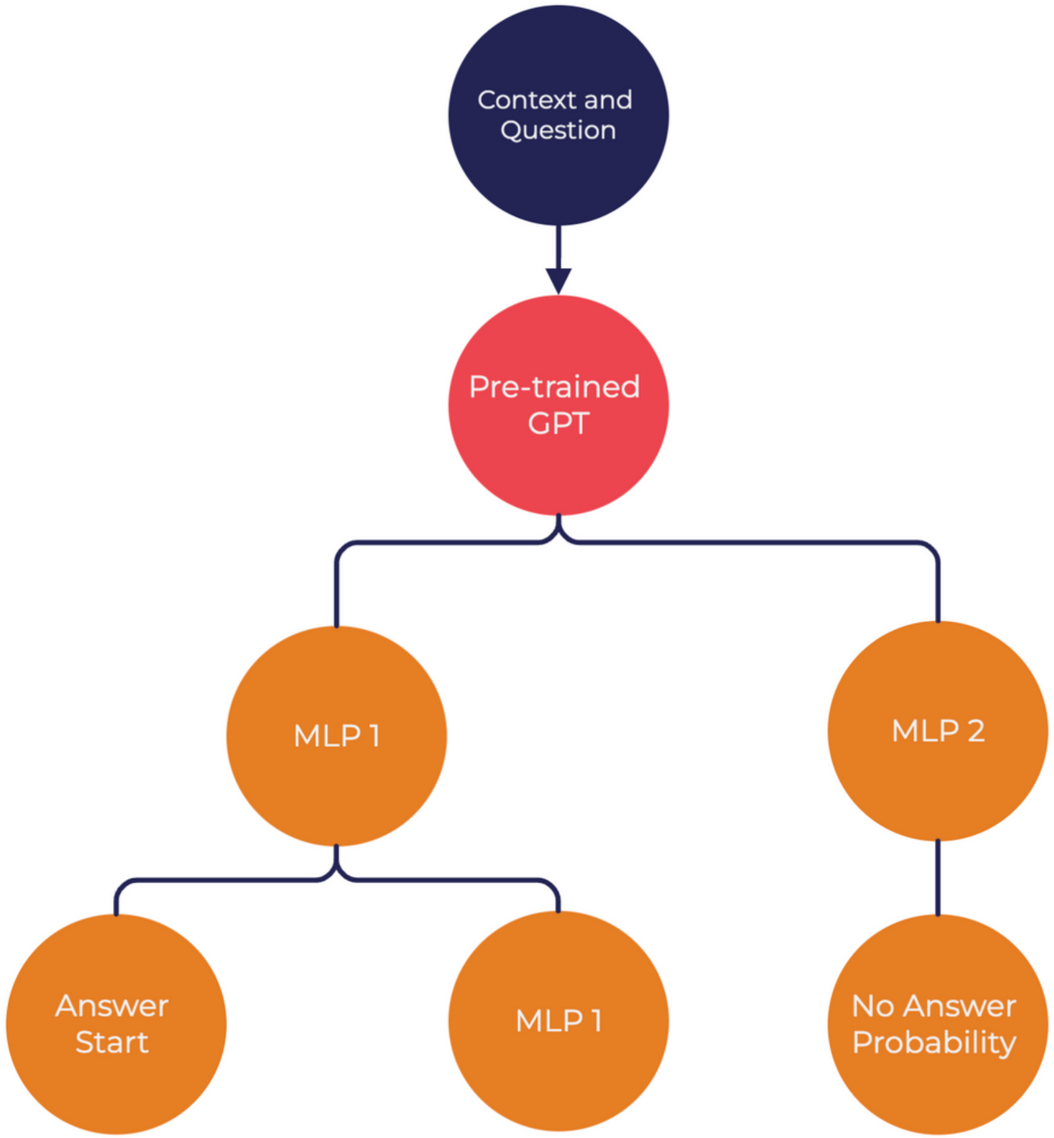
BERT-based Approach

Used different pre-trained models based on the encoder-style architecture of BERT to perform the task.

Runtime Improvements

Experimented model quantization on SpanBERT and analyzed the results.

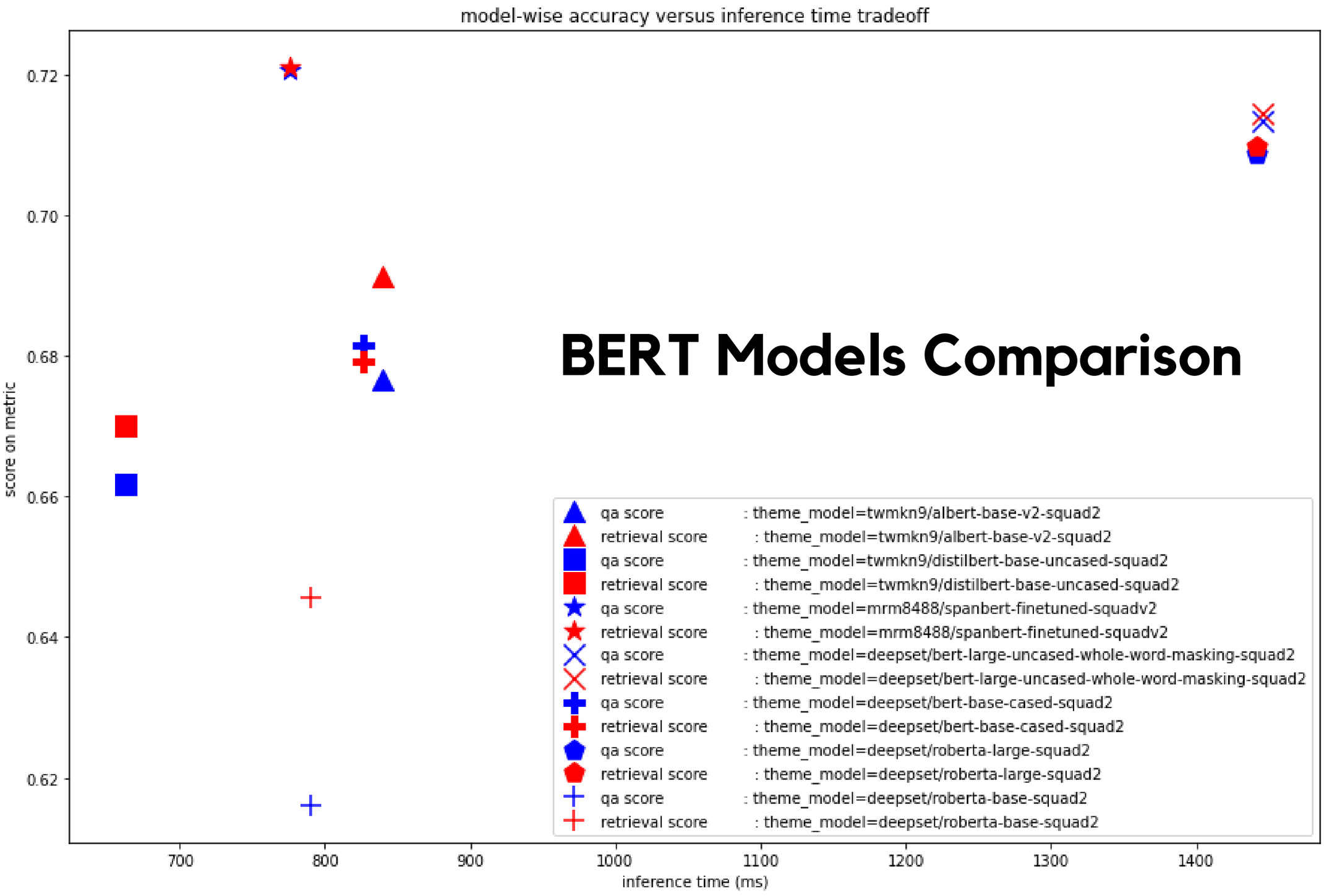
GPT-based Pipeline



Here, the GPT model is kept frozen while the perceptrons will be trained using the cross-entropy loss function in a supervised manner.

BERT-Based Approach

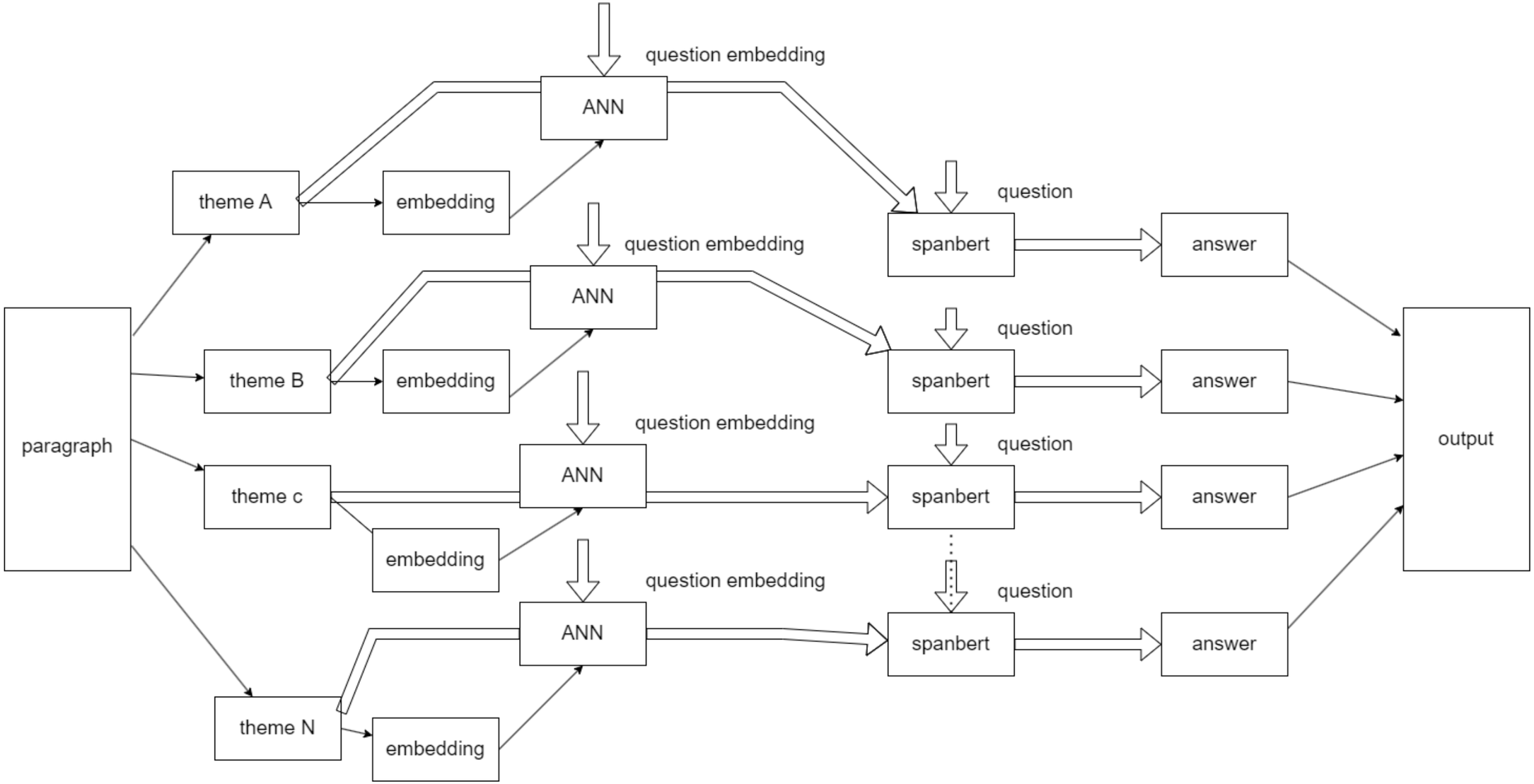
- Tried multiple models and compared their F-1 scores as well as latency for models having lower inference time
- Smaller models which were able to scan 2 paragraphs under time-limit performed poorly as compared to larger models which scanned only 1 paragraph within the time limit
- Finally selected SpanBERT due to its high F-1 score


model-wise accuracy versus inference time tradeoff
BERT Models Comparison
score on metric
inference time (ms)
0.72
0.70
0.68
0.66
0.64
0.62
700
800
900
1000
1100
1200
1300
1400
qa score : theme_model=twmkn9/albert-base-v2-squad2
retrieval score : theme_model=twmkn9/albert-base-v2-squad2
qa score : theme_model=twmkn9/distilbert-base-uncased-squad2
retrieval score : theme_model=twmkn9/distilbert-base-uncased-squad2
qa score : theme_model=mrm8488/spanbert-finetuned-squadv2
retrieval score : theme_model=mrm8488/spanbert-finetuned-squadv2
qa score : theme_model=deepset/bert-large-uncased-whole-word-masking-squad2
retrieval score : theme_model=deepset/bert-large-uncased-whole-word-masking-squad2
qa score : theme_model=deepset/bert-base-cased-squad2
retrieval score : theme_model=deepset/bert-base-cased-squad2
qa score : theme_model=deepset/roberta-large-squad2
retrieval score : theme_model=deepset/roberta-large-squad2
qa score : theme_model=deepset/roberta-base-squad2
retrieval score : theme_model=deepset/roberta-base-squad2

Runtime Improvements

- Experimented with quantization in our models but found a drop of about 5% in metrics
- Found a decrease in metrics even after scanning 2 paragraphs
- Hence, deemed it better to select other lighter models which perform similarly to the quantized models

Final Implemented Pipeline



Future Work

Use of Parallel Processing

Would improve inference time as all the cores would remain occupied all the time thereby improving the throughput of the system.

Adaption for Multilingual QA

- Translation of question to English and the retrieved answer back to the native language
- Use of multilingual models

Thank you