Hog RAGger

Set-up Process:

1) Open both .ipynb files (in src directory) actualpls_RAG and RAG_summarize, on google colaboratory.
2) Change runtime type to T4 GPU
3) Run RAG_summarize to prepare summarized.json (or just use the summarized.json available in json directory)
4) Run all cells of actualpls_RAG sequentially. You may need to restart runtime after anvil is installed.
5) Once the last cell (.wait_forever()) is called, you can access the app from the website
6) Open the website https://sociable-untidy-helmetshrike.anvil.app/
7) Submit the query to get the desired results.

Folder Structure:

```
.
|-- json
|   |-- corpus.json
|   `-- summarized.json
|-- LICENSE
|-- README.md
|-- requirements.txt
`-- src
        |-- actualpls_RAG.ipynb
        `-- RAG_summarize.ipynb
```

Overview of the Project:

EDA:

The problem demanded following inferences based on the data given:

- Query Classification
- Answer to the given query
- List of documents acting as evidence to the given answer

The following is the distribution of the types of queries given in the dataset:

Query Type	Data Quality	Quantity
Inference Query	Good	816
Comparison Query	Moderate	856
Temporal Query	Bad	583
Null Query	Good	301

The temporal query dataset was very bad, with various examples being misclassified. The major reason behind the misclassification is the similarity between comparison query and temporal query, both of which had a yes/no answer. On the other hand, inference queries had a single word answer.

Methods tried: The following metrics were noted while trying each approach:

- Inference time
- Accuracy

The following approaches were tried for various tasks:

1) Query Classification:
    - Pretrained Zero-Shot Classification Models like bart-large-mnli and roberta-large-xnli. However, the classification accuracy was terrible.
    - Bert-based-uncased fine-tuning: A bert-based-uncased model was fine-tuned on the dataset to classify the query, but the idea was dropped because the method wasn't allowed according to the rules.
    - Rule based classification: The current system uses a rule-based approach to classify incoming user queries. Queries are classified into three types: Inference, Temporal and Comparison on the basis of presence/ frequency of different keywords in the query.

2) Summarization:
    - Bart-large-cnn: Body of each document object in corpse.json was summarized using bart-large-cnn model

3) Answering:
    - Google-flan-t5-large
    - Google-flan-t5-xl
    - Llama 3.2: Used different versions of Llama 3.2 (Llama 3.2-1B, Llama 3.2-3B, Llama 3.2-3B-Instruct)
    - Google-Bert-uncased

Initial Observation / Evaluation of methods:

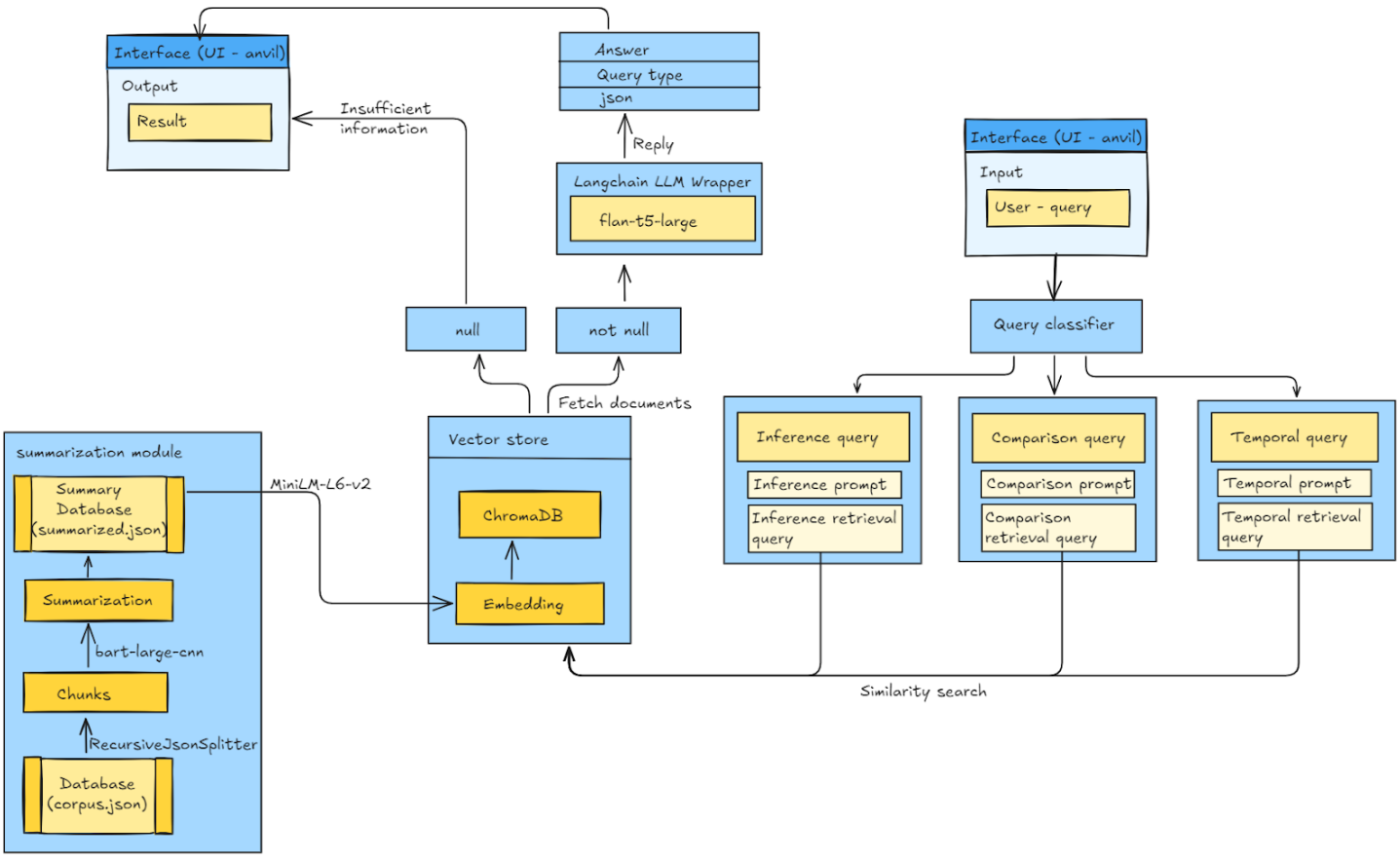
1) Directly feeding the most similar documents into answering models was ineffective due to the large size of the documents. Both BERT and Llama 3.2 models exceeded token limits when processing the documents.

2) BART-large-CNN was used to summarize the documents. While it was slower (~2 seconds per document), it produced accurate summarizations.
3) For query classification, we initially attempted to fine-tune a BERT-based uncased model on the dataset, but this resulted in overfitting. The approach was eventually abandoned due to restrictions on fine-tuning.
4) Mainly three models—Google Flan-T5-Large, Mixtral-8x7B, and Llama-3.2-3B were feasible solutions for question answering tasks. Llama-3.2-3B showed good accuracy but frequently crashed due to RAM limitations. Mixtral-8x7B was fast and accurate but required Groq API calls, which were not allowed in the competition. Although Google Flan-T5-Large didn't achieve the highest accuracy, it was the most practical option..

Final Approach:

1) After comprehensive analysis of different methods and the train-test dataset, rule-based classification was chosen for query handling, as no open-source models provided decent accuracy.
2) The BART-based CNN model was employed for document summarization, offering high accuracy and a reasonable speed. Summaries were stored in a separate file.
3) The Flan T5 Large model was used for question answering, chosen for its high speed and decent accuracy. It performed well across all query types with decent accuracy.

Flowchart:



Scope for future improvements:

1) The Anvil server setup might need to be replaced with a different web framework for production use.
2) A hybrid query classification system could be implemented with zero-shot classification deep learning models like BART MNLI reinforcing the initial prediction for better accuracy.