Retrieval Augmented Generation using GPT3.5

1. Problem Statement

The overall objective is to build an AI question answering assistant which does so using a knowledge base. This knowledge base is provided in the form of text file. The problem is solved using the methodology called Retrieval Augmented Generation (RAG) where, an LLM is used for Q&A using the context from a knowledge source.

Requirements of the tool:

1. No hallucination in answers
2. Develop methodology to evaluate the answers given by the model
3. Support multi-linguality. The user can ask questions in any language and would expect the answers to be in the language of the question
4. Support for voice based question and voice assistant providing the response

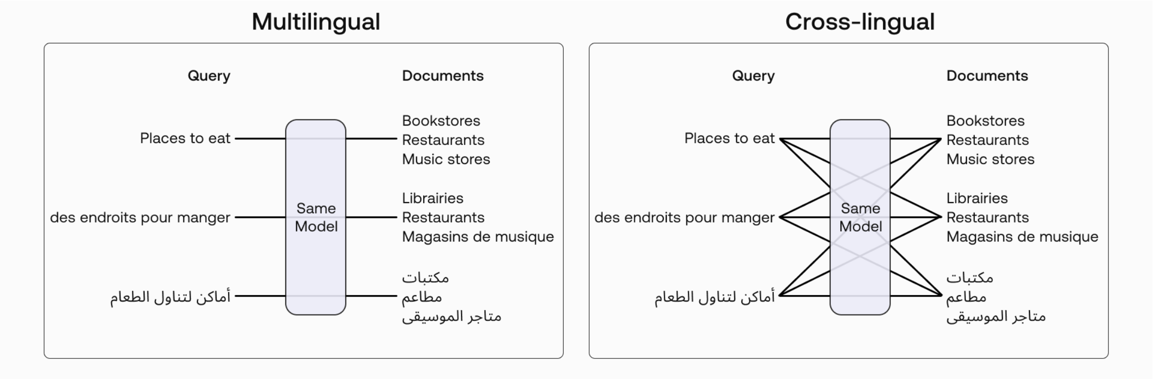
2. Approach

Some design considerations:

1. The general direction taken in this work is to use components from commercial vendors (embeddings from Cohere and LLM from OpenAI) for ease of use and rapid deployment. But langchain offers plug-and-play support for replacing any component to a custom one, if we decide to do that in future.
2. Given that the ultimate goal is for the tool to be rolled out to a large number of customers, it makes sense to save on tokens as much as possible.

   This is achieved using indexing through only 3 documents and carefully selecting the chunk_size. Whilst this is a design decision, a quick test confirms that this is enough to express the context.
3. Since the information provided in the knowledge source and the user query (typically) does not contain personal information, commercially available LLMs such as OpenAI could be used.
4. One of the requirements is to have support for multi-linguality. My assumption is, the user can use any language to type in his question and it is required that the tool can answer back in the same language. This can be achieved by using multilingual search provided by Cohere by using their embeddings.

   The image below shows how Cohere embeddings work:



The advantage is two-fold – we can add context to the knowledge base in multiple languages and the tool supports search in cross-lingual manner.

5. Another requirement was to have support for speech input. The general approach taken here was to convert the speech into text and obtain the user query. Due to limited exposure of tools, the CMU based sphinx model is used which does not require internet connection for conversion. There is scope for improvement in this area to make the conversion better. Alternatively, there are commercial vendors out there like elevenlabs or Google who provide support for this.

   Also, multilinguality is not tested with speech input and needs thorough testing before deployment.

6. The implementation of RAG is done in a Q&A manner and not as a chat since most of the questions are independent from each other and the context is not carried from one to another. But this design can be changed from Q&A to Chat since langchain offers plug and play solutions. If chat is used, we would use memory of some type (either storing the past conversation in buffer or store in a db) to store previous chat.

7. The UI is built using gradio which offers easy development and deployment. It also supports testing through API. During production, user management and session management needs to be implemented. Gradio is a chosen mode of deploying UI because it provides UI with minimal code.

8. The evaluation of the results generated is done by calculating the similarities between results and the knowledge base. The similarity score obtained represents the L2 distance between the documents and the results. This score is converted to probability (probability = 100- score). Further, these probabilities are averaged to measure the sense of relevance. This is a design decision and further testing needs to be done to see if this needs further improvement. A simple minimum score of the L2 distance might also work well given that we only need to check the most factually correct context and the result.

9. Hallucinations are prevented by specifying that the tool must obtain the results from context and not from its pretrained knowledge.

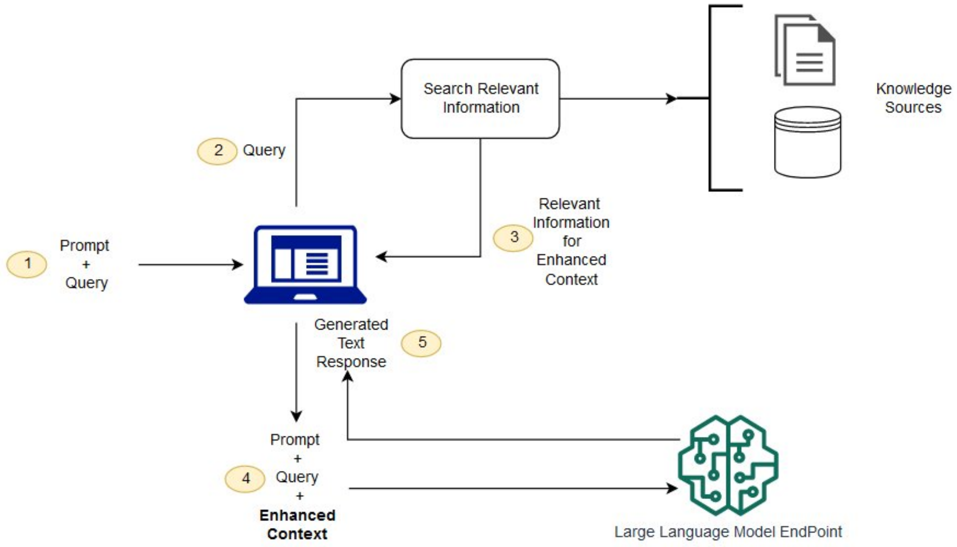
10. It is very important to have guardrails in the solution so that the LLMs are prevented from giving responses which can potentially harm the user. This is done using

constitutional chain in langchain. For now, it is required for this chain to convert the response into an ethical one. More functionalities can be added as the tool matures and the need arises.

3. Solution

Details about your solution. Illustrate performance and design with diagrams.

The RAG flow is explained in below diagram (source – link).


Search Relevant Information
Knowledge Sources
2 Query
Relevant Information for Enhanced Context
3
1 Prompt + Query
Generated Text Response
5
Prompt + Query + Enhanced Context
4
Large Language Model EndPoint

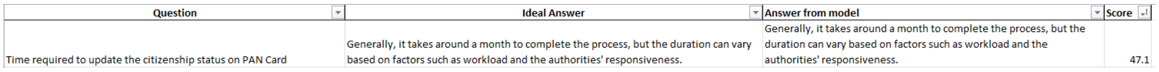
The solution contains the following components:

1. Implementation framework: Langchain
2. Knowledge source and chunking: Knowledge source is read from the txt file and MarkdownTextSplitter is used to split the text. This is mainly due to the syntax used in the text file.
3. Embeddings: Cohere multilingual embeddings are used to convert the text documents into embeddings.
4. Vector database and similarity search: FAISS is used for storing the embeddings since it is opensource.
5. LLM: 'text-davinci-003' model from OpenAI is used for this work.

5. Analysis

The responses from the model are stored in SampleQuestions_with_Responses.xlsx document. Based on the responses from the model, it can be inferred:

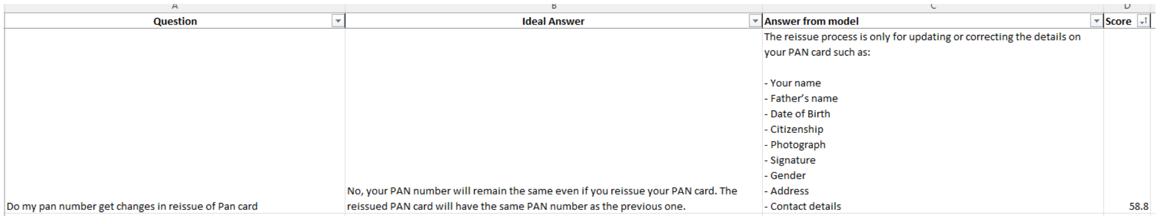
1. The model generally does a good job at generating factual responses.
2. Here is the snapshot of the answer having lowest score:



Question	Ideal Answer	Answer from model	Score
Time required to update the citizenship status on PAN Card	Generally, it takes around a month to complete the process, but the duration can vary based on factors such as workload and the authorities' responsiveness.	Generally, it takes around a month to complete the process, but the duration can vary based on factors such as workload and the authorities' responsiveness.	47.1

The score is lowest despite the fact that the ideal answer and answer from the model match very well. This can be attributed to a poor choice of evaluation metric.

3. In some responses the answer from model is longer. Prompt engineering is a key to deal with such instances.



A	B	C	D
Question	Ideal Answer	Answer from model	Score
Do my pan number get changes in reissue of Pan card	No, your PAN number will remain the same even if you reissue your PAN card. The reissued PAN card will have the same PAN number as the previous one.	The reissue process is only for updating or correcting the details on your PAN card such as: - Your name - Father's name - Date of Birth - Citizenship - Photograph - Signature - Gender - Address - Contact details	58.8

6. Future Scope

The proposed solution can be further improved by

1. Trying to pretrain on financial domain data such as rules and regulations for PAN, KYC, UIDAI to better understand the context and language
2. The method to calculate the evaluation score can be improved to check factual correctness of the figures mentioned in the response. This can be done by employing NER on the response to identify key fields such as cost, duration, names (such as SBNRI) etc. and compare them with context to see if they match.
3. For verification of the responses another LLM can be employed to verify the responses obtained. This will lead to high token usage and hence it can be done selectively to minimize cost implications.
4. Guardrails need to be in place in order to identify any PII (Personally Identifiable Information) mentioned in the prompt and must be removed before sent to model. Also a notification on screen must be shown of its potential implications. This will enable the user to trust the tool.
5. Finetuning is required in prompt engineering to make sure that the responses are inline with required length and style.