RAG Model for FAQ of ML Edge Course Content

By Afsara Maliha Hannan, Md.Golam Kausar, Showkhin Mazumder, Md. Shahriar Hossain

Introduction:

Since the advent of Large Language models, the prevalent use of these models has impacted such a wide array of commercial, educational and health sectors that it is almost difficult to imagine life without them. LLM has not only influenced activities of everyday lives but made repetitive and mundane tasks simpler to conduct. From drafting emails to creating business plans to simulating conversations regarding a plethora of topics, large language models have come a long way and surpassed the expectations of what we can achieve with text data. However, not everything in life can be this smooth without some hiccups.

Large Language models are not exactly the most consistent or credible source of information. Think of them as the friend who is knowledgeable regarding worldly affairs but can't differentiate between salt and sugar when making their morning coffee. What I am trying to convey is their lack of behavioral consistency. Much like your friend Large language models know a lot but can be highly inconsistent. So, best not to cite ChatGPT the next time you send in your term paper,

However, in most cases they nail the answer to questions, other times they regurgitate random facts from their training data. The reason why to this day LLM face this problem is because LLM were not designed to understand the information that they generate. Lets take a look at what Leibniz and Jonathan Swift had to say about this.

In the 17th century the German polymath Gottfried Leibniz published a dissertation titled "On the Combinatorial Art". Leibniz outlined a theory for automating the production of knowledge via the rule-based combination of symbols. His central argument was that all human thoughts, no matter how complex, are combinations of basic and fundamental concepts, in much the same way that sentences are combinations of words, and words combinations of letters. He believed that if he could find a way to symbolically represent these fundamental concepts and develop a method by which to combine them logically, then he would be able to generate new thoughts on demand. Or

quite possibly and frighteningly all possibly thought to have ever occurred and will occur in the history and future of humanity.

So essentially Leibniz proposed that nothing in this world will ever be original anymore. No novel idea to ponder upon because the machine will have thought of it even before you can muster the time to utter a word of your thought. What does that mean for humanity the end of the creative industry because every unicorn princess and flying pig story already exists. Well, we can't really confirm or deny that hypothesis now in the 21st century with the inception of ChatGpt, DALL-E and Sora.

But on a more positive note and less threatening to the creative process of humanity. Jonathan Swift an acclaimed author and satirist had to add to this discourse by asserting how this doctrine of pure reason was deeply flawed. Swift's point was that language is not a formal system that represents human thought, but a messy and ambiguous form of expression. To have a machine generate language requires more than having the right set of rules. It requires the ability to understand the meaning of words.

And that there is the reason why occasionally an LLM sound like they have no idea what they're saying, it's because they don't. LLMs know how words relate statistically, but not what they mean. And with that piece of information, data scientists, engineers and specialists thought it best to help the LLM model do what it does best, repeat what it is taught. The RAG(Retrieval Augmented Generation) model is different from other LLM in that we provide the model with the context or the information in advance kind of like an open book exam. The model can be broken into two main components, the retrieval part and the generation part. The retrieval part of the model retrieves information that it already has access to and then the second task of the architecture is to pass the retrieved information into a generation model now all it has to do is generate a human-like response without too much work required.

To provide a general overview, the RAG model is an AI framework for retrieving facts from an external database to ground large language models (LLMs) on the most accurate, up-to-date information. It leverages external knowledge sources, typically knowledge graphs or other knowledge bases, such as Wikipedia, to augment the generation capabilities of these models. Implementing RAG in an LLM-based question answering system has two main benefits: It ensures that the model has access to the most current, reliable facts, and that users have access to the model's sources, ensuring that its claims can be checked for accuracy and ultimately trusted. This

integration enables the model to incorporate additional external information during text generation, thereby improving its accuracy and applicability while ensuring access to the latest reliable information.

The motivation of this project is to create a question answering RAG model that will answer all the queries pertaining to the lectures delivered in the ML edge class. The dataset used in this project comes from the texts in all the PDFs provided in the ML edge Google classroom. The document has been manually partitioned and the text is sectioned according to possible questions that can be asked from the lecture context. There are 190 questions and 203 answers which is extracted from two csv documents ‘queries.csv’ and ‘answers.csv’ respectively.

Methodology:

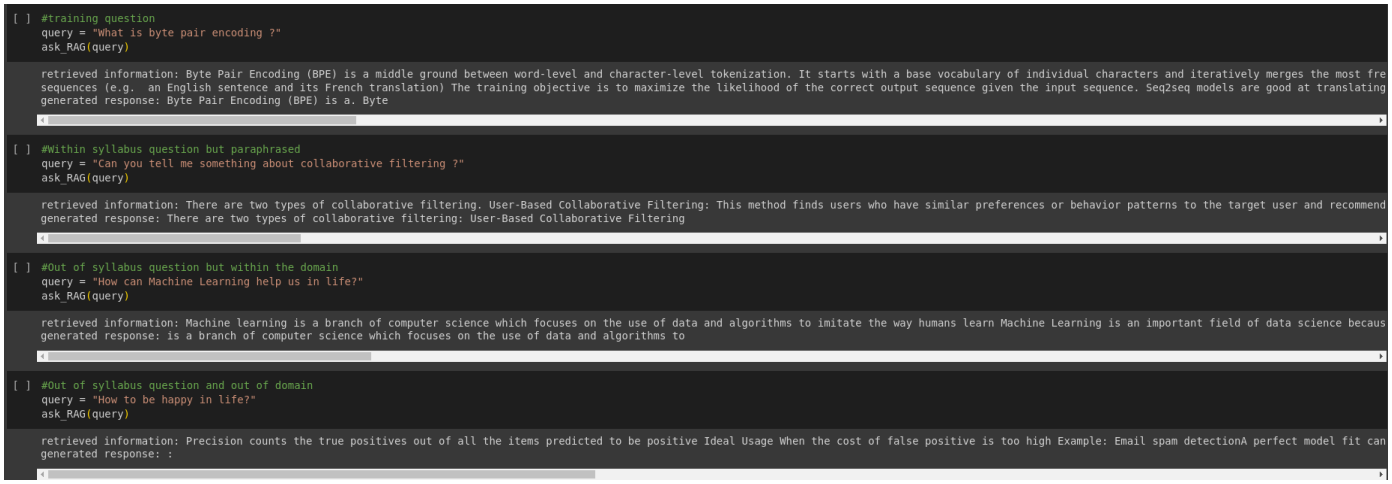
In this current project, the RAG model has been built on the foundation of two LLM models. The retrieval part is conducted with the help of a pre-trained ColBert(Contextualized Late Interaction over BERT) model. ColBert is a ranking model that improves upon the BERT model for efficient retrieval. The major task conducted in the retrieval process is word embeddings and conducting similarity tests in order to compare the queries with the available information. ColBert introduces a late interaction architecture that independently encodes the query and the document using BERT and then integrates a powerful interaction step to model their similarity tests. The advantages of using a ColBert model is that it has the ability to pre-compute the documents offline which speeds up the query processing.

In the code, the word embedding part of the model is conducted in the Indexing segment where a module by the name of ‘Indexing’ inside the ‘colbert’ library is used. In the indexing phase, the indexer processes a large collection of documents and each document is represented as a dense vector using techniques like Bidirectional Encoder Representations from Transformers. This vector representation captures the semantic meaning of the document. The similarity tests is conducted with the help of the ‘Searcher’ module. The searcher takes the user query as an input and then retrieves relevant documents from the already indexed collection. It encodes the query into a dense vector representation using the same rule as was done on the documents. After which it compares the representation with the representations of the indexed documents. The ColBert model contributes to a lower computational cost. The model also optimizes itself with a pruning-friendly interaction mechanism that facilitates vector-similarity indexes for end-to-end retrieval directly from a large collection of document.

Once the retrieval part of the model is complete, the documents retrieved will now be passed onto the generation part of the project. The generation sector is conducted using the T5-base (Text-To-Text Transfer Transformer) model. The T5 is a transformer-based language model designed by Google. The base version is a medium sized variant of the model architecture. The T5 model is trained to map input text to output text, enabling it to perform a wide range of natural language processing tasks using a unified framework.

In the case of this project, a pre-trained T5-base model has been fine-tuned on the dataset pertaining to all the lectures from the ML Edge class. For the sake of simplicity and due to lack of computational power, the model was not rigorously trained. With only 2 epochs in the training process the model demonstrated a training loss of 2.277 and a validation loss of 1.49 in the first epoch. Whilst the second epoch had a training loss of 1.84 and a validation loss of 1.28 which indubitably is an improvement in model training.

In terms of the model output, the retrieval part of the model works much better. The top 3 searches are extracted and passed onto the generation model. Where the retrieved data in most cases are the top most relevant text information necessary to generate a comprehensive response. However, the generated response from the T5-base model is not the most satisfactory. Here are some examples of the information retrieved and the response generated from queries.



Since the response from the generative model was not satisfactory, the model also has the option of generating the response from an OpenAI GPT-3.5 API using a Langchain framework. The generated responses are much better in this case.

Conclusion

As a preliminary step to understand the working of a RAG model, this project can be used as an introduction for students to understand how the RAG framework works. Predominantly, a RAG model contains two parts the retrieval and the generation of response. The model responsible for retrieval of information will embed and then extract from the database on the basis of similarity scores. Therefore, the database from which the information is extracted is often a large corpus with topic specific accurate information.

However, there is more work to be done to improve the model performance. This can be optimized by improving the quality of the dataset and segment the information pertaining to each question into smaller chunks to feed the generative model. Since the ColBERT model is already working excellently in retrieving the information, it would not do much good to fine-tune the model on such a small dataset. However, the generative model of T5-base needs to be fine-tuned with better parameters.

Bibliography

Aryani, A. (2024, January 19). *A Brief Introduction to Retrieval Augmented Generation (RAG)*. Medium. https://medium.com/@amiraryani/a-brief-introduction-to-retrieval-augmented-generation-rag-4bd6e50da532

Bhardwaj, V. (2022, August 19). *ColBERT: A complete guide - Varun Bhardwaj - Medium*. Medium. https://medium.com/@varun030403/colbert-a-complete-guide-1552468335ae

Schwartz, O. (2021, September 30). *In the 17th Century, Leibniz Dreamed of a Machine That Could Calculate Ideas*. IEEE Spectrum. https://spectrum.ieee.org/in-the-17th-century-leibniz-dreamed-of-a-machine-that-could-calculate-ideas

Zhang, Z. (2022, December 2). *ColBERT — A Late Interaction Model For Semantic Search*. Medium. https://medium.com/@zz1409/colbert-a-late-interaction-model-for-semantic-search-da00f052d30e