Retrieval-Augmented Generation : A Basic Overview

The Problem with LLMs

Large Language Models are trained on immense text datasets. LLMs are pretty equipped at finding patterns, comprehending the grammar and understanding the context. We can quantify the “knowledge” of an LLM by looking at the size of the dataset it had been trained on. In simple words, the more detailed the training data is, the better is the Language model’s performance in terms of context specific answers.

But what if the training data of the model gets outdated, incomplete or has insufficient material to give a specific answer in response to the query the user has given?

This is where the Language Model starts to “Hallucinate”. A model is said to hallucinate when it generates wrong, nonsensical or fabricated information in response to a user’s query. Main cause is due to the very mathematical and statistical methods of models to interpret and relate the information, which means lacking “intuitive” understanding of the real world or some specific topic related to the query.Hallucinations thus decrease the reliability of LLM outputs, because nobody wants a model which just “makes up stuff”, or just forgets to share the most crucial information, because it can not understand the proper context or does not have the knowledge specific to the user’s request.

Along with this, following are some of the other problems

- Supplying old information (based on what it was trained on)
- Extracting precise information from proprietary documents of an organisation.
- Lost in the Middle phenomenon - The information which is near the middle of the context window is often overlooked or forgotten by the models, while processing large and lengthy data.

These are the problems which make large organisations reluctant to use Language Models for their work.
For example - For a legal research assistant, being unaware of the recent developments and amendments in the laws could lead to insufficient or even completely incorrect outputs.

The Solution

RAG (Retrieval Augmented Generation) emerges as a critical technique for enhancing the reliability and accuracy of LLMs in such real world applications.

RAG significantly reduces hallucinations by augmenting the user's prompt with all the other relevant documents related to the query so that the Language Model is equipped with up to date, detailed and correct information about the subject. This leads to enhanced understanding of the topic and thus better, more accurate responses.

Data Retrieval

Data retrieval is the very crucial step where the system identifies and retrieves the data and information which are most relevant to the user's query. Data retrieval is done using two prominent approaches ie: Lexical and Semantic retrieval.

1. Lexical Retrieval

- The traditional approach.
- Based on exact word matches and the frequency analysis.
- Common methods are TF-IDF and BM25

1.1 Term Frequency - Inverse Document frequency (TF-IDF)

This is a numerical statistic which is used to determine the importance of a word in the document corpus. It characterises a word based on two basic metrics, first - how often does the word appear in that individual document and second - how infrequently (less often) the word appears in the entire corpus.

- Process -
  - Term Frequency (TF) -
    - Term Frequency means the relative frequency of the term (token) within a particular document.
    - Calculated by taking the ratio of the frequency of a term to the total number of terms in the document.
    - ###Math Formula###
    - Aim is to determine the importance of that term within the document.
  - Inverse Document Frequency (IDF) -
    - IDF score tells about the rarity of the term in the document corpus.
    - Calculated by taking the log of the ratio of total number of documents in the corpus to the number of documents containing that term.
    - ###Math Formula###
    - Aim is to penalize the ubiquity of a term within the corpus.

- Problems -
    - The method does not take into account the saturation of term frequency.
    - Lack of any ability to differentiate between rare and common words except the IDF scaling.

1.2 Best Matching 25 (BM25)

BM25 also adds methods which take the term frequency saturation in mind and also normalizes the length of document, resulting in better scoring.

- Advantages (over TF-IDF) -
    - After a certain threshold occurrence of the term does not increase its importance in the document, avoids over emphasizing of overly generic terms.
    - TF-IDF tends to favour longer documents because they have more terms, but BM25 normalizes the document length such that shorter documents are also considered if they contain the terms relevant to the query. Thus eliminating the bias towards document length.
    - Does not use a fixed formula like TF-IDF, thus BM25 is more adaptable.

1.3 Advantages of Lexical retrieval

- **Efficiency and Speed -** Because lexical retrieval uses basic string operations and does not require much of complex mathematical calculations, it is fast and demands less computational power.
- **No Training data Required -** For exact keyword mapping, no training data is required. Thus useful for languages(or domains) for which labelled data is unobtainable.
- Due to the above advantages, Lexical retrieval can be better for scenarios where the query is well explained and demands the information relevant to particular keywords.

1.4 Some Limitations

- **No Semantic Understanding -** Lexical retrieval methods have no scope of semantic understanding ie: they can not differentiate between synonyms and paraphrases.
- Vulnerability towards typographical errors and misspelling.

2. Semantic Retrieval

2.1 The Process-

- Vector Encoding -
    - Machine learning models transform both the query and the document into vectors in a shared vector space.
    - These models are trained on large data sets so that they can encode the information properly.
- Semantic Matching -
    - Vectors in the query are compared using metrics like Cosine Similarity.
    - All the relevant documents found are then augmented with the query and then supplied to the LLM.

2.2 Advantages -

- Capturing meaning and Context - Now the documents are retrieved with a broader context, along with exact keyword matches, now the model also fetches relevant documents on the basis of meaning and possible interpretations.
- Enables more complex queries - The queries can be more complex and nuanced, the user is now not required to use explicit terms related to the domain, but a general description of the subject leads to retrieval of the documents required for proper response.

2.3 Disadvantages -

- Retrieval now based on an ML model, it faces a few challenges now, that are -
- Significant demand of computational resources,
- Possibility of biases in the retrieval system because it is based on its own training data.
    - Along with all that, updating the document embeddings for ever changing contents can be very complex and also resource intensive.

3. Hybrid Retrieval and RRF

- Hybrid Retrieval utilizes the strengths of both the lexical and semantic retrieval methods.

3.1 Reciprocal Rank Fusion (RRF) -

- RRF merges the ranking of terms, obtained by both the lexical method(BM25 scores) and the semantic method. Multiple retrieval methods can also be incorporated for a more robust system.
- Process -
    - Each token receives a score based on their rankings from the different retrieval models.
    - Each score thus gives an idea of how important the term is, with varying metrics on which the term's importance is weighed by the different models.
    - The scores obtained are then combined together using the following formula.
        - ###Math###
    - The output of this formula serves as the combined relevance score of the term in the document corpus.
    - The final output contains the first K documents arranged according to the decreasing order of their combined scores.
- Advantages -
    - Understands Term Relationship - The model now understands how different terms are related to each other in a document.
    - Better Recall - The terms which could have been missed by either of the earlier retrieval methods individually, are being looked at by the hybrid retriever.
        - For example - Semantic retrieval models often miss those terms or phrases on which they haven't been trained upon. Using hybrid retrieval thus utilizes lexical retrieval methods.

RAG for Knowledge-Intensive NLP Tasks

As we have clearly seen till now, RAG has proved to be a very helpful system which enhances the performance of Large Language Models, by providing them with appropriate relevant documents retrieved from a vast dynamic pool of data along with

the user query. RAG prevents the model from hallucinating by giving it proper context, specific references and updated data about the real world.

Referring to the Research paper "**RAG for Knowledge-Intensive NLP Tasks"**
(proper citation kaise karna hai nahi pata)

In this paper, we look into the complete process of RAG in detail -

1. **Models**
    a. **RAG-Sequence(One Passage per Query)-**
        i. Only a single passage is generated which is augmented with the query and then passed through the LLM
        ii. This passage is retrieved using the neural retrievers which encode the query, compare it with the pre encoded vector database of the corpus and retrieve the relevant information using metrics like cosine similarities.
        iii. Now this retrieved document is treated as a Latent variable which is further marginalised to obtain response from the model.
        iv. ###math###

    b. **RAG-Token (One passage per Token)-**
        i. In contrast to Rag sequence, Rag token uses a more dynamic interplay with the corpus, where for every term of the query, retriever finds all the related relevant documents.
        ii. This repeated retrieval thus results in more nuanced responses, because the model now has access to a wider range of documents which cover a broader range of context. In other words, the model becomes even more flexible.
        iii. However this flexibility comes at a cost of potential risk of inconsistency. The cause of this inconsistency would be the potential inclusion of unrelated, out of context documents from the corpus because of per token based retrieval from the query.
        iv. ###math###

2. **Retriever: DPR**
    Dense Passage Retrieval (DPR) -
    a. In DPR, we use a pretrained bi-encoder to initialize the retriever and index the documents.

b. The Query encoding is done by a neural encoder like BERTd. From this step we obtain the dense representation of our query. Which is then compared with the dense document corpus.
c. Probability to obtain some document z for a token x in the query is proportional to the dot product of the dense representation of the document and the query token.
d. ###math###
e. Finding the best match for the query or a particular token (depending on the type of rag model used), is nothing more than a Maximum Inner Product Search (MIPS) problem.

Note: Dense representation is more detailed and nuanced than one hot encoding, in this kind of representation, every token corresponds to a vector in a multidimensional vector space. Data is “encoded” into such dense representation so that it can be semantically compared with each other.

3. Generator

a. The generator can be any large language model, which would receive an augmented input of the query and the relevant documents retrieved from the corpus, by the RAG model.
b. In the research paper, BART-large is used, which is a pre-trained model with 400M parameters.

4. Training

a.