Retrieval Augmented Generation

Chapter 1

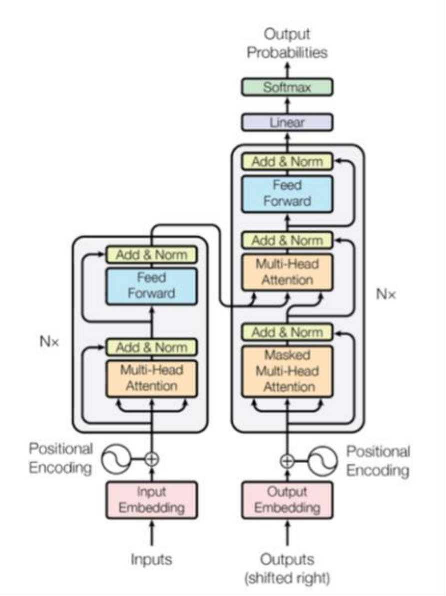
The technique of retrieving relevant information from an external source, augmenting the input to the LLM with that external information, thereby enabling the LLM to generate an accurate response is called Retrieval Augmented Generation.

As the name implies, Retrieval Augmented Generation, in three steps -

1. *Retrieves relevant information from a data source external to the LLMs*
2. *Augments the input to the LLM with that external information*
3. *Then, the LLM Generates a more accurate result.*

Large Language Models (LLMs), in particular, and generative artificial intelligence (AI) in general are versatile technologies with a wide range of uses. In general, LLMs can be viewed as a prediction model for next tokens, or more informally, next words. These are machine learning models that have discovered statistical patterns to mimic human-like language skills after learning from enormous datasets of human-generated text.

A straightforward network architecture based on attention mechanisms called "transformers" has enabled large language models. Before transformers were developed, complicated recurrent neural networks (RNNs) or convolutional neural networks (CNNs) in an encoder-decoder configuration were used to perform tasks like language synthesis.



Transformers architecture in an encoder-decoder model

If our use case is in a domain where the vocabulary and the syntax of the language is very different from commonly spoken language then chances are that the available LLMs may not yield optimal results. Domains like healthcare prescription data where the vocabulary is very specific or legal domain where the meaning of words is very different from common language may require collection of domain specific data and training a language model from scratch.

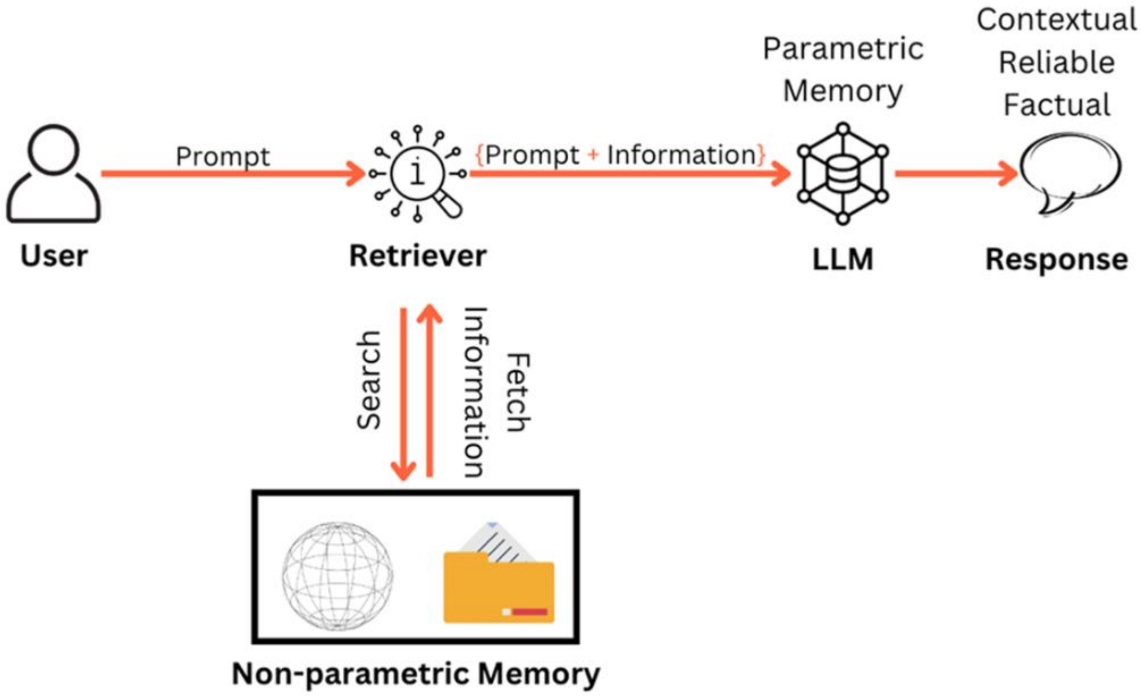
The mathematical form of all machine learning models, including LLMs, is $y=f(x)$, where x is the training data's characteristics and y represents the model. Consider the following equation: $y = w + b1x1 + b2x2 + b3x3 +\dots. + bnXn$. In this case, the values w, $b1$, $b2$,…bn are those that the model learns or modifies during training. We refer to these numbers as model parameters.

Researchers and practitioners have identified several elements of a prompt that help an LLM provide better answers. As an illustration, It has been shown that defining a "role" for the LLM, such as "You are a software engineer who is an expert in Python" or "You are a marketer who excels at creating digital marketing campaigns," improves the caliber of responses. One of the best ways to direct the LLM responses has been found to be providing "examples" within the prompt. Another name for this is Few Shot Prompting. Additionally, it has been noted that providing precise and comprehensive instructions facilitates rapid adherence.

*The limitations experienced by LLMs are: Knowledge cut-off, Hallucinations, Knowledge limitation and these limitations are addressed in a*ccording to Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (https://arxiv.org/abs/2005.11401), RAG models integrate pre-trained "parametric" and "non-parametric" memory to generate language.

An LLM's capacity to remember data that it has been trained on depends only on its parameters. Thus, it can be claimed that the parameters of LLMs contain factual information. The parametric memory is the term used to describe this internal memory found in the LLM. There are limits to this parametric memory. It is a function of the data used to train the LLM and is dependent on the quantity of parameters. The word "non-parametric" refers to this data that is not part of the LLM but can be supplied to it. The "non-parametric" memory of the system is formed if we are able to collect data from outside sources whenever we want and utilize it in conjunction with the LLM.

RAG also increases user confidence in the LLM responses as a result of resolving the problem of limited parametric memory.The additional data helps the LLM produce contextually relevant responses, giving users a sense of greater assurance. For instance, users can be sure that the LLM will produce responses regarding a certain company's items from the sources they have provided and not from other sources if the non-parametric memory has information about those products. Because the information is being retrieved from a recognized source, it is not only context aware but also able to be referenced in the answer. Because users can choose to validate the information from the source, this increases the reliability of the responses. With contextual awareness, the tendency of LLM responses to be factually inaccurate is greatly reduced. The LLMs hallucinate less in RAG enabled systems.



The most popular use cases of RAG are : Search engines, personalized marketing experience, Real-time event commentary, conversational agents, Document question answering system, Virtual agents.