# Retrieval Augmented Generation (RAG)

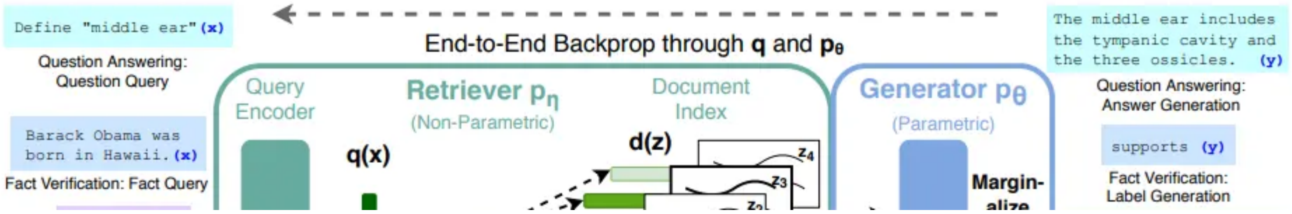
General-purpose language models can be fine-tuned to achieve several common tasks such as sentiment analysis and named entity recognition. These tasks generally don't require additional background knowledge.

For more complex and knowledge-intensive tasks, it's possible to build a language model-based system that accesses external knowledge sources to complete tasks. This enables more factual consistency, improves reliability of the generated responses, and helps to mitigate the problem of "hallucination".

Meta AI researchers introduced a method called Retrieval Augmented Generation (RAG) to address such knowledge-intensive tasks. RAG combines an information retrieval component with a text generator model. RAG can be fine-tuned and its internal knowledge can be modified in an efficient manner and without needing retraining of the entire model.

RAG takes an input and retrieves a set of relevant/supporting documents given a source (e.g., Wikipedia). The documents are concatenated as context with the original input prompt and fed to the text generator which produces the final output. This makes RAG adaptive for situations where facts could evolve over time. This is very useful as LLMs's parametric knowledge is static. RAG allows language models to bypass retraining, enabling access to the latest information for generating reliable outputs via retrieval-based generation.

Lewis et al., (2021) proposed a general-purpose fine-tuning recipe for RAG. A pre-trained seq2seq model is used as the parametric memory and a dense vector index of Wikipedia is used as non-parametric memory (accessed using a neural pre-trained retriever). Below is a overview of how the approach works:


Define "middle ear" (x)
Question Answering: Question Query
Barack Obama was born in Hawaii. (x)
Fact Verification: Fact Query
End-to-End Backprop through q and pθ
Query Encoder
Retriever pη
(Non-Parametric)
Document Index
q(x)
d(z)
z4
z3
z2
Generator pθ
(Parametric)
Margin-alize
The middle ear includes the tympanic cavity and the three ossicles. (y)
Question Answering: Answer Generation
supports (y)
Fact Verification: Label Generation

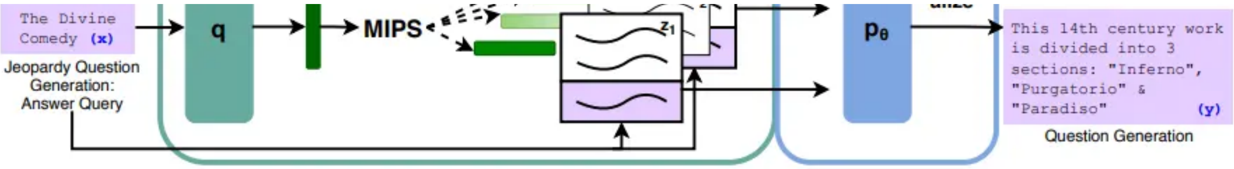

The Divine Comedy (x)
Jeopardy Question Generation: Answer Query
q
MIPS
$z_1$
$p_\theta$
This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso" (y)
Question Generation

Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query $x$, we use Maximum Inner Product Search (MIPS) to find the top-K documents $z_i$. For final prediction $y$, we treat $z$ as a latent variable and marginalize over seq2seq predictions given different documents.

Image Source: Lewis et el. (2021)

RAG performs strong on several benchmarks such as Natural Questions, WebQuestions, and CuratedTrec. RAG generates responses that are more factual, specific, and diverse when tested on MS-MARCO and Jeopardy questions. RAG also improves results on FEVER fact verification.

This shows the potential of RAG as a viable option for enhancing outputs of language models in knowledge-intensive tasks.

More recently, these retriever-based approaches have become more popular and are combined with popular LLMs like ChatGPT to improve capabilities and factual consistency.

## RAG Use Case: Generating Friendly ML Paper Titles

Below, we have prepared a notebook tutorial showcasing the use of open-source LLMs to build a RAG system for generating short and concise machine learning paper titles:

Getting Started with RAG

# References

- Retrieval-Augmented Generation for Large Language Models: A Survey (Dec 2023)
- Retrieval Augmented Generation: Streamlining the creation of intelligent natural language processing models (Sep 2020)

Last updated on November 18, 2024