W&B Fully Connected > Articles > LLM

A Gentle Introduction to Retrieval Augmented Generation (RAG)

In this article, we will learn about Retrieval Augmented Generation (RAG) and how it helps pre-trained LLM models to generate more specific, diverse and factual responses.

Piyush Thakur

Created on August 26 | Last edited on February 28

Large language models (LLMs) have undeniably propelled NLP task-specific architectures by excelling at storing and learning factual information from extensive data. However, their performance falters when faced with knowledge-intensive tasks due to inherent limitations.

LLMs struggle to access and manipulate knowledge effectively, as they cannot readily expand or update their memory. Moreover, they may produce erroneous outputs known as "hallucinations" and often fail to provide clear insights into their predictions.

To solve the limitations of LLMs, Retrieval Augmented Generation (RAG) has gained significant attention and is redefining the way we approach text generation tasks.

In exploring how and why, here's what we'll be covering:

What is Retrieval Augmented Generation (RAG)?
How Does RAG Work?

The RAG Framework
    RAG Components
    The RAG Retriever
    The RAG Ranker
    The RAG Generator
RAG Techniques and Models
The Benefits of Retrieval Augmented Generation
Applications of RAG
Challenges and Future Directions of RAG
Getting Started with RAG
Conclusion

What is Retrieval Augmented Generation (RAG)?

RAG is an AI framework that retrieves facts from an external knowledge base and helps pre-trained large language models generate more accurate, up-to-date information and reduce hallucinations.

Retrieval Augmented Generation has garnered increasing attention due to its ability to overcome several limitations associated with traditional text generation models. While generative models, like OpenAI's GPT, have demonstrated remarkable capabilities in generating coherent and contextually relevant text, they often fall short in tasks requiring specific, factual information or fine-grained control over content. By combining the strengths of both retrieval and generation, RAG models address these limitations and pave the way for more versatile and effective text generation.

In an era where NLP applications are ubiquitous, ranging from chatbots and content generation to question-answering systems and language translation, Retrieval Augmented Generation techniques offer a powerful solution to elevate the quality and reliability of these applications. Whether it's generating informative responses to user queries, crafting content that blends creativity with accuracy, or producing multilingual translations with precision, RAG models have begun to play a pivotal role.

How Does RAG Work?

RAG can be likened to a detective and storyteller duo. Imagine you are trying to solve a complex mystery. The detective's role is to gather clues, evidence, and historical records related to the case. Once the detective has compiled this information, the storyteller designs a compelling narrative that weaves together the facts and presents a coherent story. In the context of AI, RAG operates similarly.

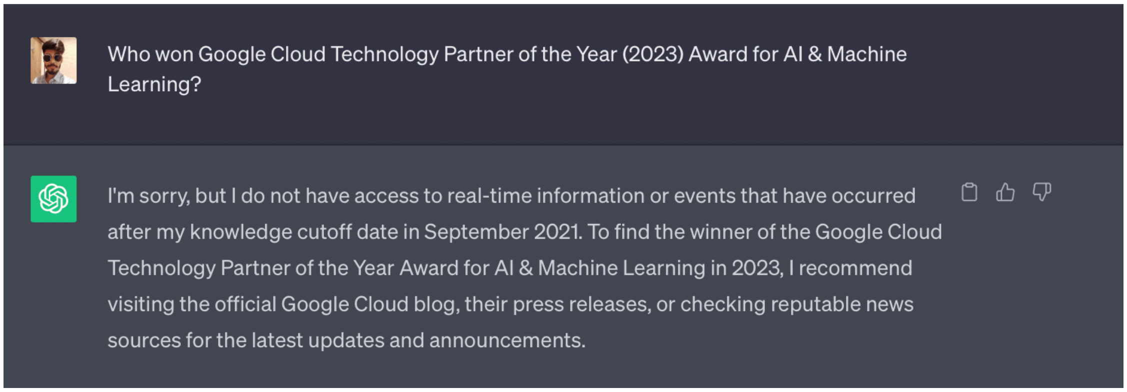
The **Retriever Component** acts as the detective, scouring databases, documents, and knowledge sources for relevant information and evidence. It compiles a comprehensive set of facts and data points.

The **Generator Component** assumes the role of the storyteller. Taking the collected information and transforming it into a coherent and engaging narrative, presenting a clear and detailed account of the mystery, much like a detective novel author.

This analogy illustrates how RAG combines the investigative power of retrieval with the creative skills of text generation to produce informative and engaging content, just as our detective and storyteller work together to unravel and present a compelling mystery.

Here's another example to understand this better.

As illustrated in the screenshot below, I initially posed a question to ChatGPT that lay beyond the scope of its database, thus preventing it from providing a specific answer. In response, ChatGPT informed us that its knowledge is limited to information available up to its knowledge cutoff date in September 2021. This scenario highlights the inherent limitation of ChatGPT's ability to provide information on events or developments occurring after that date.

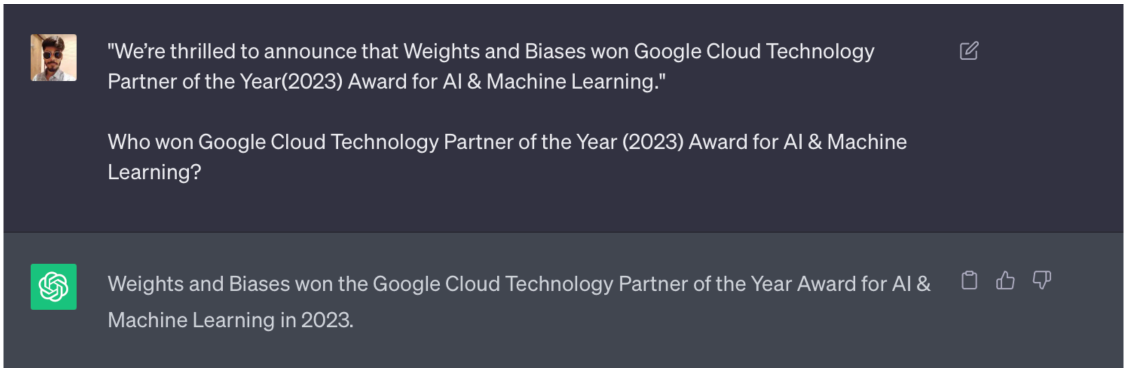


Who won Google Cloud Technology Partner of the Year (2023) Award for AI & Machine Learning?

I'm sorry, but I do not have access to real-time information or events that have occurred after my knowledge cutoff date in September 2021. To find the winner of the Google Cloud Technology Partner of the Year Award for AI & Machine Learning in 2023, I recommend visiting the official Google Cloud blog, their press releases, or checking reputable news sources for the latest updates and announcements.

Screenshot of ChatGPT

As illustrated in the screenshot below, I augmented the original question with additional context to elicit a more accurate response from ChatGPT. While the answer provided seemed plausible, it lacked concrete evidence or a verifiable source to substantiate its correctness. This underscores an important aspect of AI-generated responses – they often lack the ability to provide factual proof or cite sources, leaving users to exercise caution and independent verification when necessary.



"We're thrilled to announce that Weights and Biases won Google Cloud Technology Partner of the Year(2023) Award for AI & Machine Learning."

Who won Google Cloud Technology Partner of the Year (2023) Award for AI & Machine Learning?

Weights and Biases won the Google Cloud Technology Partner of the Year Award for AI & Machine Learning in 2023.

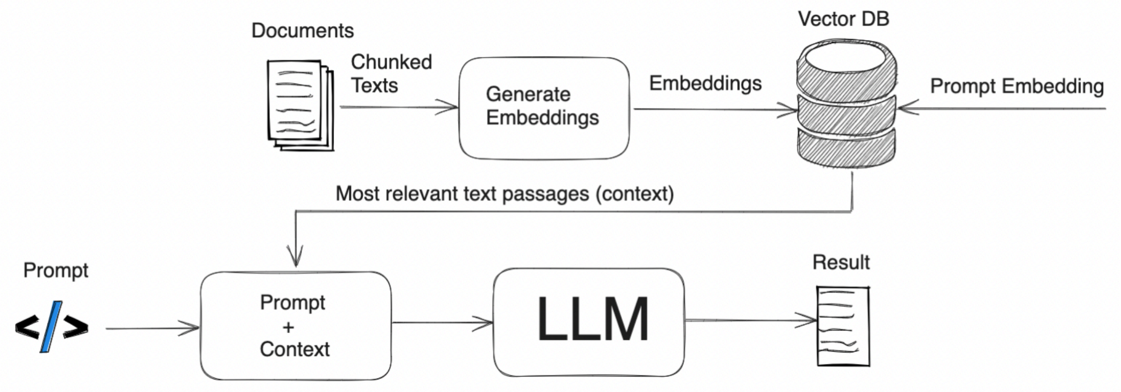
Screenshot of ChatGPT

From these two scenarios, we can draw two key observations:

- The large language model (LLM) used here operates with outdated information, as it lacks access to the most recent and reliable facts beyond its knowledge cutoff date.
- Furthermore, the responses provided by the LLM do not reference its sources, which means that its claims cannot be independently verified for accuracy or relied upon with complete trust. This highlights the importance of independent fact-checking and critical assessment when using AI-generated information.

To address these limitations, RAG emerges as a solution. Let's delve into RAG's framework to understand how it mitigates these challenges.

The RAG Framework



Source

Let's go through each part here:

- **Prompt** - At the onset, the user provides a prompt outlining their expectations for the response.
- **Contextual Search** - This pivotal step involves augmenting the original prompt with external contextual information. An external program responsible for searching and retrieving data from various sources comes into play. This process may encompass querying a relational database, conducting keyword-based searches within indexed documents, or even invoking APIs to fetch data from remote or external sources.
- **Prompt augmentation** - Following the contextual search, the additional information retrieved is seamlessly integrated into the original user prompt. This augmentation enriches the user's query with factual data, enhancing its depth and relevance.
- **Inference** - With this augmented and context-enriched prompt in hand, the Language Model (LLM) comes into play. The LLM, now armed with both the original user query and the supplementary context, significantly enhances its accuracy. It can tap into factual data sources to provide more precise and contextually relevant responses.
- **Response** - The LLM formulates the response, incorporating factually correct information. This response is then relayed back,

ensuring that the user receives accurate and reliable answers to their queries.

In essence, RAG's framework leverages external contextual information to improve the accuracy and informativeness of responses, addressing the limitations of outdated knowledge and the inability to verify information that may be associated with traditional language models.

RAG Components

Essentially, Retrieval Augmented Generation models are comprises of three components:

- Retriever
- Ranker
- Generator

Let's understand each one in detail.

The RAG Retriever

The RAG retriever component is responsible for the initial step of retrieving relevant information from external knowledge sources. It uses retrieval techniques such as keyword-based search, document retrieval, or structured database queries to fetch pertinent data.

The retriever can employ pre-built indexes, search algorithms, or APIs to access various knowledge sources, including databases, documents, websites, and more.

Its primary goal is to compile a set of contextually relevant information that can be used to enrich the user's query.

The RAG Ranker

The RAG ranker component refines the retrieved information by assessing its relevance and importance. It assigns scores or ranks to the retrieved data points, helping prioritize the most relevant ones.

Rankers use various algorithms, such as text similarity metrics, context-aware ranking models, or machine learning techniques, to evaluate the quality of the retrieved content.

This step ensures the most pertinent information is presented to the generator for content generation.

The RAG Generator

The RAG generator component is responsible for taking the retrieved and ranked information, along with the user's original query, and generating the final response or output.

It employs generative models, such as transformer-based models (e.g., GPT, BERT), to craft human-like text that is contextually relevant, coherent, and informative.

The generator ensures that the response aligns with the user's query and incorporates the factual knowledge retrieved from external sources.

RAG Techniques and Models

We treat the retrieved document as a latent variable to train both the retriever and generator together. We introduce two models for this purpose:

- **RAG-Sequence -** In this model, the same retrieved document is used to predict each token in the target sequence. It maintains consistency by relying on a single document throughout the generation process.
- **RAG-Token -** In the RAG-Token approach, different tokens in the target sequence can be predicted based on different documents. This allows for more flexibility as each token can benefit from the most relevant context.

To summarize, RAG models blend input sequences, retrieved documents, and generation to produce text. The retriever finds relevant documents, and the generator uses this context to predict each token. We have two approaches, RAG-Sequence and RAG-

Token, to incorporate retrieved documents differently during text generation. These models enable us to generate coherent and contextually relevant text based on user queries and retrieved information.

The Benefits of Retrieval Augmented Generation

Retrieval Augmented Generation offers a range of benefits in the field of NLP and text generation:

- **Improved Accuracy** - RAG models provide factually accurate information by leveraging knowledge from external sources. This makes them valuable in applications where precision and reliability are paramount, such as question-answering and content generation for educational purposes.
- **Contextual Relevance** - RAG enhances the contextual relevance of the generated text. By incorporating external context, RAG-generated responses are more likely to align with the user's query or context, providing more meaningful and contextually appropriate answers.
- **Enhanced Coherence** - The integration of external context ensures that RAG-generated content maintains logical flow and coherence. This is particularly valuable when generating longer pieces of text or narratives.
- **Versatility** - RAG models are versatile and can adapt to a wide range of tasks and query types. They are not limited to specific domains and can provide relevant information across various subjects.
- **Efficiency** - RAG models can efficiently access and retrieve information from large knowledge sources, saving time compared to manual searches. This efficiency is especially valuable in applications where quick responses are essential, such as chatbots.
- **Content Summarization** - RAG is useful for summarizing lengthy documents or articles by selecting the most relevant information and generating concise summaries. This aids in information digestion and simplifies content consumption.

- **Customization** - RAG systems can be fine-tuned and customized for specific domains or applications. This allows organizations to tailor the models to their unique needs and requirements.
- **Multilingual Capabilities** - RAG models can access and generate content in multiple languages, making them suitable for international applications, translation tasks, and cross-cultural communication.
- **Decision Support** - RAG can assist in decision-making processes by providing well-researched, fact-based information that supports informed choices in various fields, including healthcare, finance, and legal.
- **Reduce Manual Effort** - RAG reduces the need for manual research and information retrieval, saving human effort and resources. This is particularly valuable in scenarios where large volumes of data need to be processed.
- **Innovative Applications** - RAG opens doors to innovative NLP applications, including intelligent chatbots, virtual assistants, automated content generation, and more, enhancing user experiences and productivity.

Applications of RAG

RAG finds applications in various domains and industries, leveraging its ability to combine retrieval-based and generative techniques to enhance text generation and information retrieval. Here are some notable applications of RAG:

- **Question Answering Systems** - RAG is particularly valuable in question-answering applications. It can retrieve and generate precise and contextually relevant answers to user queries, making it suitable for virtual assistants, FAQs, and expert systems.
- **Chatbots and Virtual Assistants** - RAG-powered chatbots can provide more accurate and informative responses to user inquiries. They excel in natural language interactions, making them ideal for customer support, information retrieval, and conversational AI.
- **Content Summarization** - RAG can be employed to summarize lengthy documents, articles, or reports by selecting the most

salient information and generating concise summaries. This is useful for content curation and information digestion.

- **Information Retrieval** - RAG can enhance traditional information retrieval systems by providing more contextually relevant and coherent results. It improves the precision and recall of search engines, making it valuable in research and knowledge management.
- **Content Generation** - RAG is used to generate content for various purposes, including news articles, reports, product descriptions, and more. It ensures that the generated content is factually accurate and contextually relevant.
- **Educational Tools** - RAG can assist in creating educational materials by generating explanations, study guides, and tutorials. It ensures that the content is informative and aligned with the educational context.
- **Legal Research** - In the legal domain, RAG can be applied to retrieve case law, statutes, and legal opinions. It helps lawyers and legal professionals access relevant legal information efficiently.
- **Healthcare Decision Support** - RAG can assist healthcare professionals in decision-making by providing up-to-date medical information, research findings, and treatment guidelines. It aids in evidence-based medicine.
- **Financial Analysis** - RAG models can generate financial reports, market summaries, and investment recommendations based on real-time data and financial databases, assisting analysts and investors.
- **Cross-Lingual Applications** - RAG's multilingual capabilities are beneficial for translation tasks, cross-cultural communication, and information retrieval in multiple languages.
- **Content Moderation** - RAG can assist in content moderation on online platforms by identifying and generating responses to user-generated content that violates guidelines or policies.
- **Knowledge Bases and Expert Systems** - RAG can be used to update and expand knowledge bases in real-time, ensuring that expert systems have access to the most current information.
- **Search Engine Optimization (SEO)** - RAG can assist in generating SEO-friendly content by selecting relevant keywords and optimizing content for search engine rankings.
- **Data Extraction** - RAG can be used to extract structured information from unstructured text data, facilitating data mining

and analysis tasks.
- **Historical Data Analysis** - RAG can help historians and researchers analyze historical texts, documents, and archives by providing contextually relevant information and generating historical narratives.

These applications highlight the versatility and utility of RAG in various fields, where the combination of retrieval and generation capabilities significantly enhances text-based tasks and information retrieval processes.

Challenges and Future Directions of RAG

While RAG is a powerful approach in NLP, it still comes with its own challenges. Here are some of the key challenges associated with RAG:

- **Handling Diverse Query Types -** RAG models need to be versatile enough to handle a wide range of query types, from straightforward factual questions to more complex, nuanced queries. Adapting the retrieval and generation components to suit this diversity can be challenging.
- **Balancing Retrieval and Generation -** Striking the right balance between retrieval and generation is crucial. Over-relying on retrieval may lead to responses that lack creativity or context, while excessive generation may result in less factual or relevant answers.
- **Scaling to Large Datasets -** As knowledge bases and data sources continue to grow, RAG models must scale efficiently. Handling massive datasets without sacrificing response times and accuracy is a technical challenge.
- **Evaluation Metrics -** Assessing the performance of RAG models can be complex. Traditional metrics may not fully capture the quality of responses, especially when factual accuracy and contextual relevance are critical.
- **Ethical Considerations -** RAG models can inadvertently generate content that may be biased, offensive, or harmful. Ensuring

responsible use and mitigating ethical concerns in content generation is an ongoing challenge.
- **Limited Real-time Information -** RAG models are often based on static knowledge sources, which means they may not provide accurate information for rapidly changing real-time events or developments.
- **Cost and Resource Intensiveness -** Implementing RAG systems, particularly with large-scale knowledge bases, can be resource-intensive in terms of computation, storage, and data preprocessing.

Getting Started with RAG

RAG is an exciting field in NLP that combines information retrieval with text generation. Whether you're a researcher, developer, or enthusiast looking to explore RAG techniques, here's a comprehensive guide to help you get started:

Recommended Resources

- **Paper** - "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" by Lewis et al. (2020).
- **Tools** - Langchain, Pinecone, and LlamaIndex are streamlining the process of retrieval augmentation, making it more efficient and user-friendly.
- **Datasets** - There are various datasets available such as TriviaQA which is designed for question answering and retrieval tasks. It's a good starting point for experimenting with RAG models. Natural Questions dataset contains real user questions for information retrieval. It's valuable for training and evaluating RAG systems. MS MARCO is a large-scale dataset for machine reading comprehension and document ranking. It offers diverse challenges for RAG research.
- **Libraries and Frameworks** - Huggingface Transformers has the RAG model that leverages external documents (like Wikipedia) to augment its knowledge and achieve state of the art results on knowledge-intensive tasks.

Conclusion

RAG is a groundbreaking approach in NLP that overcomes the limitations of traditional language models. By seamlessly combining retrieval and generation, RAG improves the accuracy and context relevance of text generation. It addresses the challenges faced by large pre-trained language models, offering benefits like enhanced coherence and versatility. RAG finds applications in various domains, and while it presents challenges, ongoing research is working to overcome them. For those interested in RAG, a plethora of resources and tools are available. RAG is poised to shape the future of text generation and has a significant role in NLP advancements.

Tags: Articles, Framework / Integration, LLM

Created with ❤️ on Weights & Biases.

https://wandb.ai/cosmo3769/RAG/reports/A-Gentle-Introduction-to-Retrieval-Augmented-Generation-RAG---Vmlldzo1MjM4Mjk1

Made with Weights & Biases. **Sign up** or **log in** to create reports like this one.

Iterate on AI agents and models faster. Try Weights & Biases today.

SIGN UP

TRY W&B NOW

Weights & Biases

Get weekly updates with the latest ML news.

Subscribe

Copyright ©2025 Weights & Biases. All rights reserved.