



# What is RAG (Retrieval-Augmented Generation)?



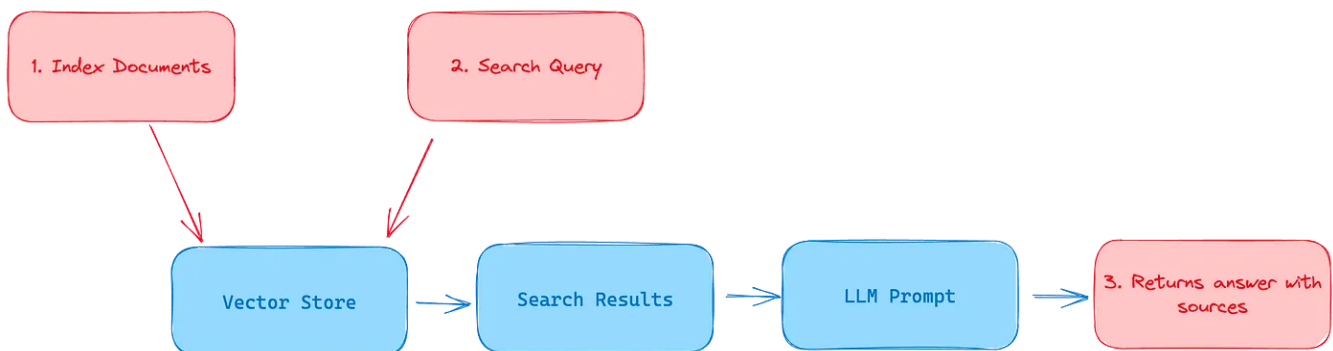
Jacky · Follow

3 min read · Jun 24



Listen

Retrieval-augmented generation is a technique used in natural language processing that combines the power of both retrieval-based models and generative models to enhance the quality and relevance of generated text.



The most primitive architecture for retrieval-augmented generation.

To understand retrieval-augmented generation, let's break it down into its two main components: retrieval models and generative models.

1. **Retrieval models:** These models are designed to retrieve relevant information from a given set of documents or a knowledge base. They typically use techniques like information retrieval or semantic search techniques to identify the most relevant pieces of information based on a given query. Retrieval-based models excel at finding accurate and specific information but lack the ability to generate creative or novel content.
2. **Generative models:** Generative models, on the other hand, are designed to generate new content based on a given prompt or context. These LLMs (now

explained **all** over the internet) use a large amount of training data to learn the patterns and structures of natural language. Generative models can generate creative and coherent text, but they may struggle with factual accuracy or relevance to a specific context.

Now, retrieval-augmented generation combines these two approaches to overcome their individual limitations. In this framework, a retrieval-based model is used to retrieve relevant information from a knowledge base or a set of documents based on a given query or context. The retrieved information is then used as input or additional context for the generative model.

By incorporating the retrieved information, the generative model can leverage the accuracy and specificity of the retrieval-based model to produce more relevant and accurate text. It helps the generative model to stay grounded in the available knowledge and generate text that aligns with the retrieved information.

### *About retrieval models*

Retrieve models are a type of language model that focus on *finding* relevant information from a dataset, in response to a given query. These models can benefit from vast stores of knowledge and are usually trained to produce meaningful and context-specific results. The most common examples of retrieval models:

Retrieval models are generally designed to find and rank relevant pieces of information from a dataset in response to a query. Here are some examples of popular retrieval models and algorithms:

1. **Neural Network Embeddings:** Neural network embeddings (Such as OpenAI's embeddings or Cohere's embeddings) ranks documents based on their similarity in the vector space.
2. **BM25 (Best Match 25):** A widely used text retrieval model based on probabilistic information retrieval theory. It ranks documents based on term frequencies and inverse document frequencies, considering both the relevance and rarity of terms within a corpus.
3. **TF-IDF (Term Frequency — Inverse Document Frequency):** A classic information retrieval model that measures the importance of a term within a document relative to the whole corpus. It combines term frequency (how often a term

appears in a document) and inverse document frequency (how rare the term is in a corpus) to rank documents in relevance.

4. Hybrid Search: a combination of the above methodologies with different weightings.
5. There are a few other methods but such as LDA but they're not particularly powerful by themselves as of yet.

### *Applications*

Retrieval-augmented generation has several applications. For example, in question-answering systems, the retrieval-based model can find relevant passages or documents containing the answer, and the generative model can then generate a concise and coherent response based on that information. In content generation tasks, such as summarization or story writing, the retrieval-based model can provide relevant facts or context, which the generative model can use to create more informative and engaging content.

In summary, retrieval-augmented generation combines the strengths of retrieval-based models and generative models to improve the quality and relevance of generated text. By leveraging the retrieval-based model's ability to find accurate information and the generative model's ability to produce creative text, this approach enables more robust and contextually grounded language generation systems.

### *Building your own RAG engine*

There are a few solutions out there where you can test building your own RAG engine (I will be writing and sharing my experiences on these soon!).

- If you are interested in an interesting open-source solution, I recommend checking out **haystack**
- **Langchain** also offers this but their solution right now is quite inflexible and it's not clear how results can be improved if they are bad

[Data Science](#)[Large Language Models](#)[Llm](#)[Rag](#)[Artificial Intelligence](#)



Follow

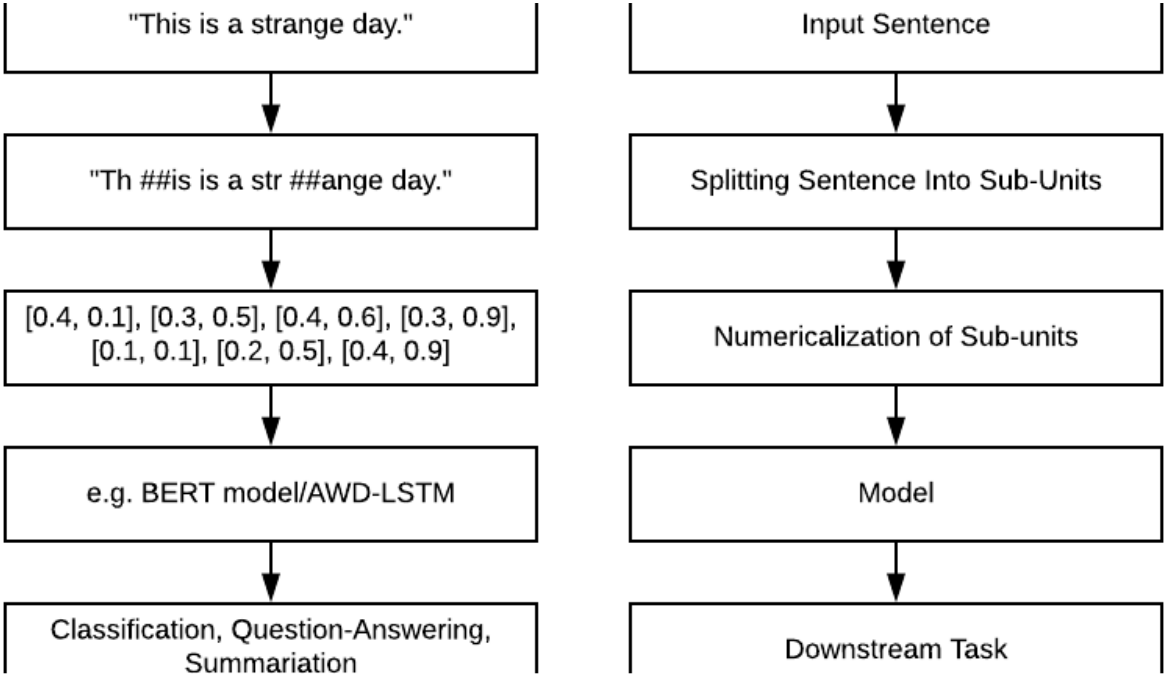
# Written by Jacky

68 Followers

Scientist/Engineer

---

## More from Jacky



 Jacky

## Understanding SentencePiece ([Under][Standing]\_Sentence][Piece])

As SentencePiece is used in many cutting-edge NLP models, I decided to go into depth to explore what SentencePiece is about and understand...

11 min read · May 21, 2020

 208

 4



 Jacky

 **DeepEval—Synthetic Data, Bulk Review, Custom Metric Logging and**


DeepEval v0.14 Update

3 min read · Sep 12

 2    1



	3	0.9	0.081
	4	0.9	0.0729
	5	0.9	0.06561
	6	0.9	0.059049
	7	0.9	0.0531441
	8	0.9	0.04782969
	9	0.9	0.043046721
	10	0.9	0.0387420489

 Jacky in Artificial Intelligence in Plain English

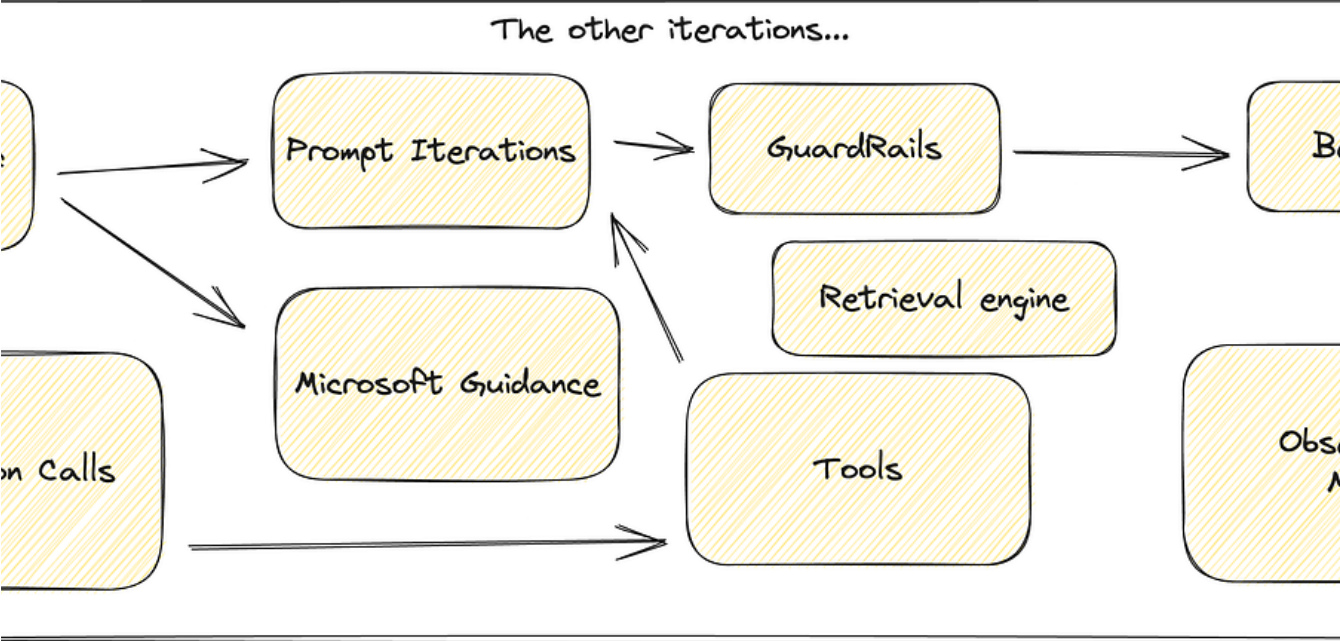
Comparing Top-K Rankings Statistically

For data scientists, ML Engineers, ML developers, search engine enthusiasts.

4 min read · Jan 4, 2021

 5   





Jacky

# Stop Eye-Balling If Prompts Work, How To Test LLMs

The Problem

5 min read · Sep 13

24




See all from Jacky

## Recommended from Medium





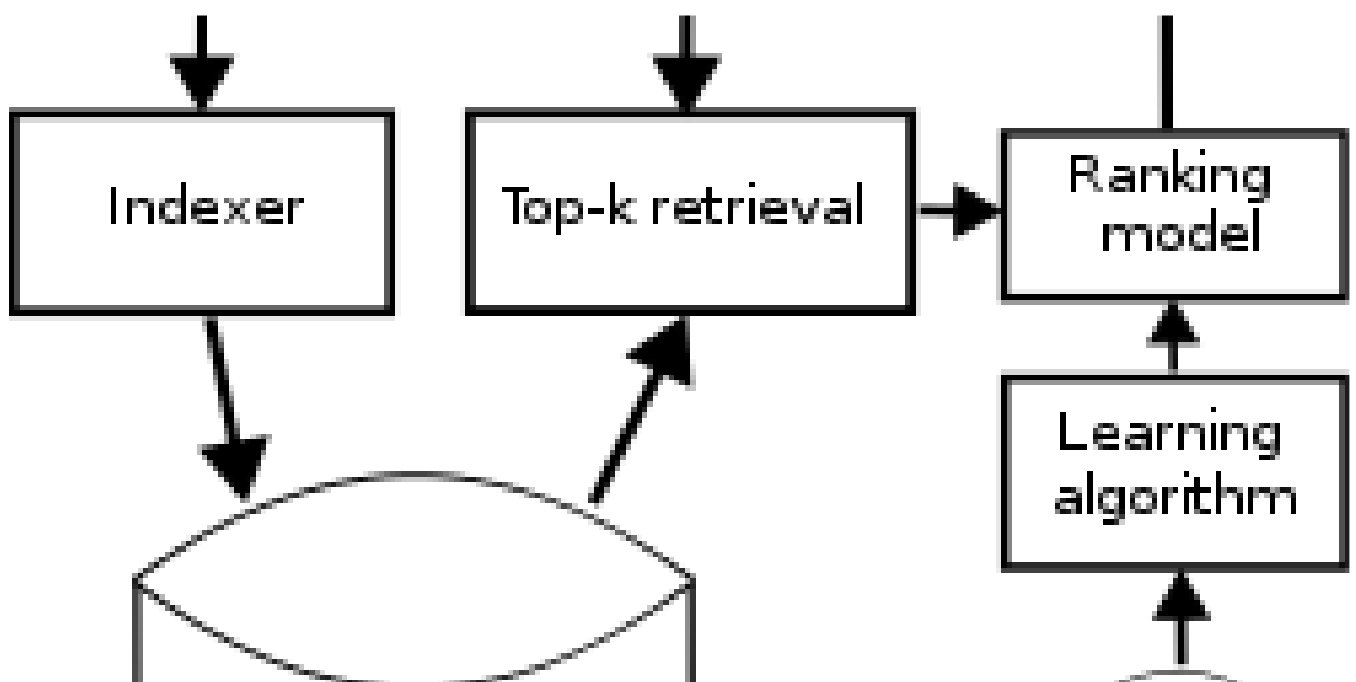
 Amogh Agastya in Better Programming

## Harnessing Retrieval Augmented Generation With Langchain

Implementing RAG using Langchain

19 min read · Sep 21

 359  4



 Shivam Solanki  in Towards Generative AI

## Improving RAG (Retrieval Augmented Generation) Answer Quality with Re-ranker



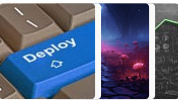
## Implementing the Re-ranker algorithm in the RAG pipeline

5 min read · Aug 4

 94     2



### Lists



**Predictive Modeling w/ Python**  
20 stories · 516 saves



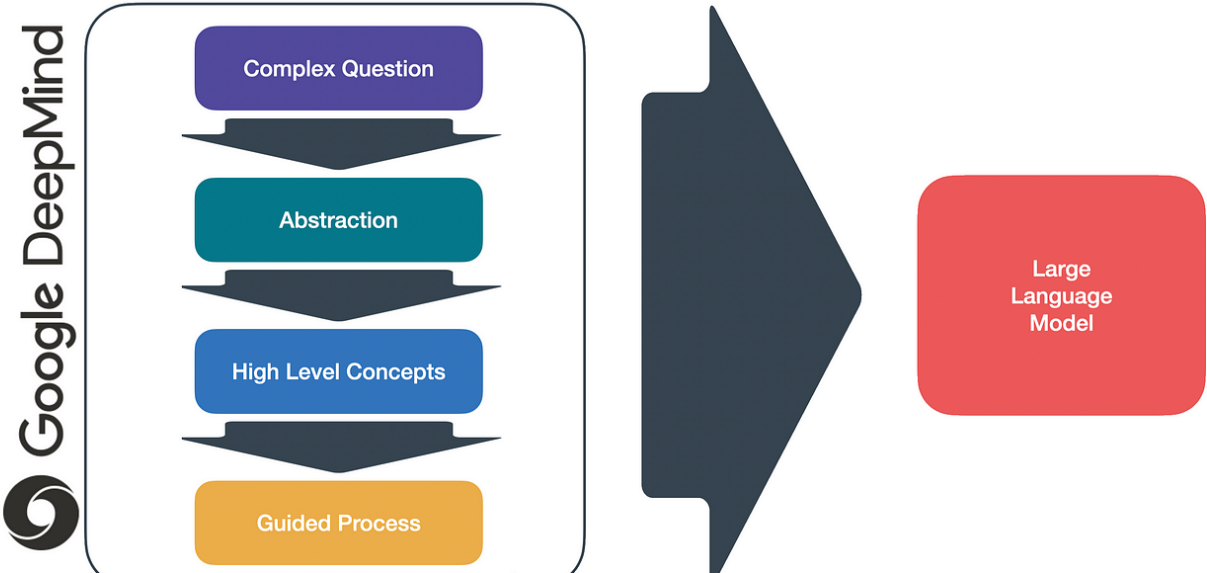
**Natural Language Processing**  
739 stories · 331 saves




**AI Regulation**  
6 stories · 159 saves



**ChatGPT prompts**  
27 stories · 530 saves



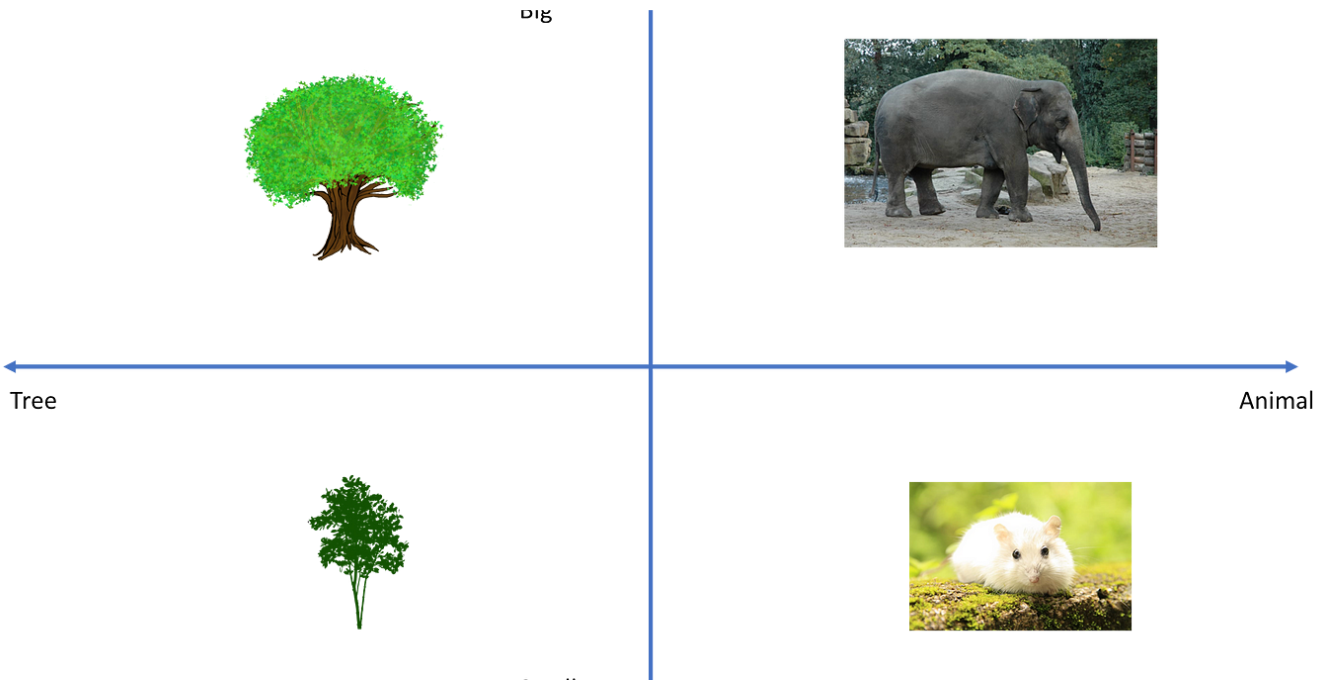
 Cobus Greyling

### A New Prompt Engineering Technique Has Been Introduced Called Step-Back Prompting

Step-Back Prompting is a prompting technique enabling LLMs to perform abstractions, derive high-level concepts & first principles from...

5 min read · Oct 12

 771  9



 Skanda Vivek in Towards Data Science

## Build Industry-Specific LLMs Using Retrieval Augmented Generation

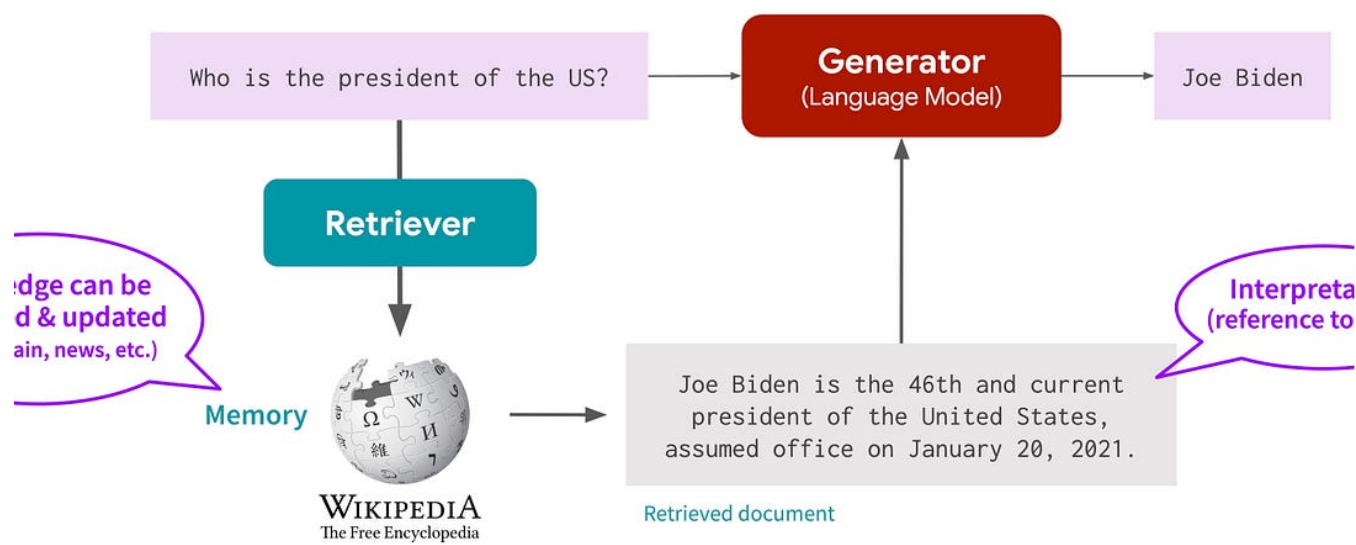
Organizations are in a race to adopt Large Language Models. Let’s dive into how you can build industry-specific LLMs Through RAG


★ · 10 min read · May 31

 458  10



# Retrieval augmentation



 Akriti Upadhyay in Accredian

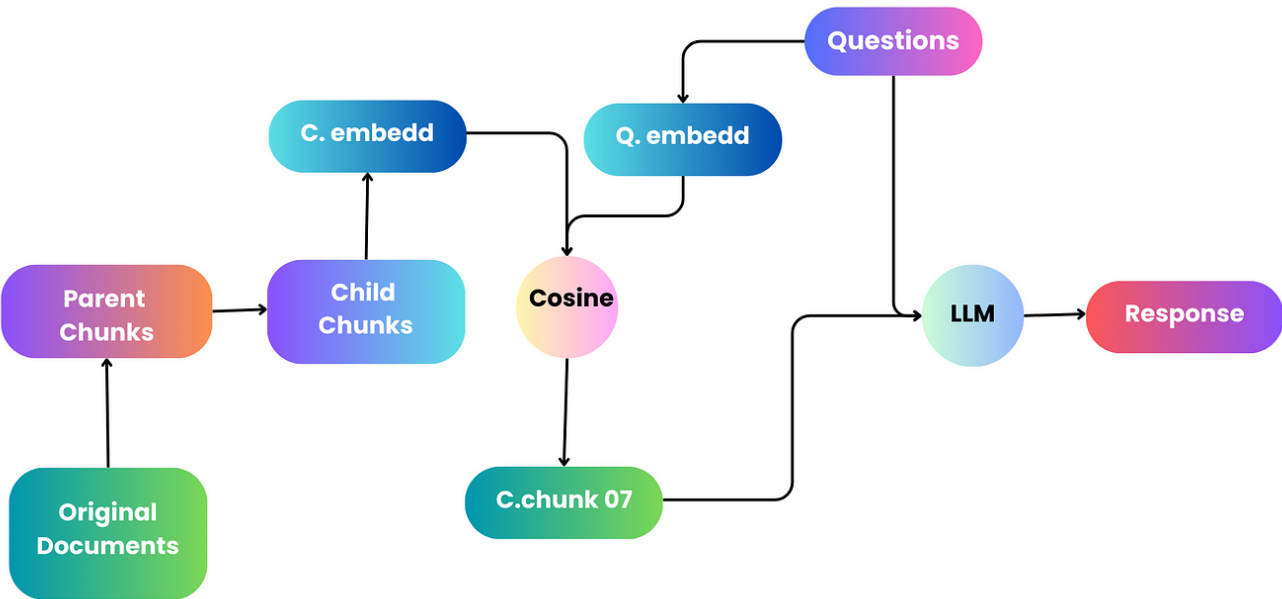
## Efficient Information Retrieval with RAG Workflow


Improving Search and Summarization using Retrieval Augmented Generation

6 min read · Oct 9

 213     3





 azhar in ai insights cobet

## RAG and Parent Document Retrievers: Making Sense of Complex Contexts with Code

# Introduction

6 min read · Oct 14



44



2



See more recommendations