Retrieval Augmented Generation (RAG) Cheatsheet

Overview

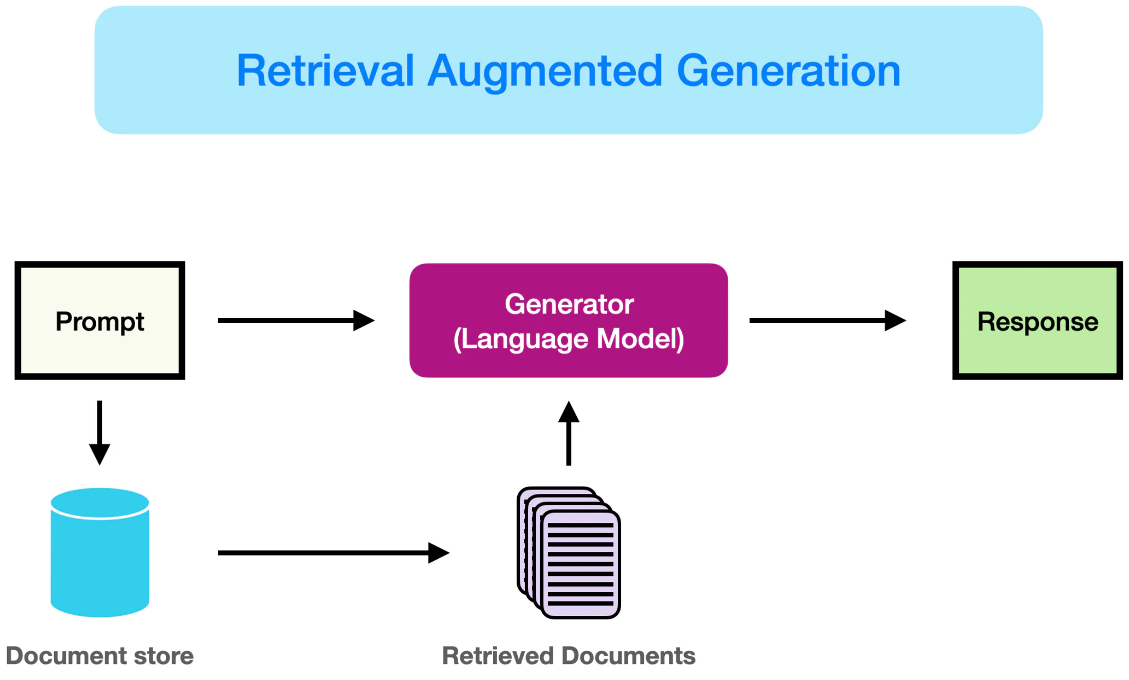


Figure 1: alt text

- **RAG**: A solution to mitigate challenges in LLMs such as domain knowledge gaps, factuality issues, and hallucination by augmenting LLMs with external knowledge.
- **Use Cases**: Knowledge-intensive scenarios or domain-specific applications that require continually updating knowledge.
- **Advantage**: No need for retraining the LLM for task-specific applications.

Key Findings from "Retrieval-Augmented Generation for Large Language Models: A Survey"

- **Focus Areas**: Existing approaches, state-of-the-art RAG, evaluation, applications, and technologies surrounding retrieval, generation, and augmentation techniques.

Introduction to RAG

- **Definition**: RAG retrieves a set of relevant/supporting documents given a source (e.g., Wikipedia), concatenates them as context with the original input prompt, and feeds them to the text generator to produce the final output.
- **Benefits**: Enhances the accuracy, controllability, and relevancy of the LLM's response. Reduces issues of hallucination or performance when addressing problems in a highly evolving environment.

Evolution of RAG

- **Current Approaches**: Combining the strengths of RAG and powerful fine-tuned models like ChatGPT and Mixtral.

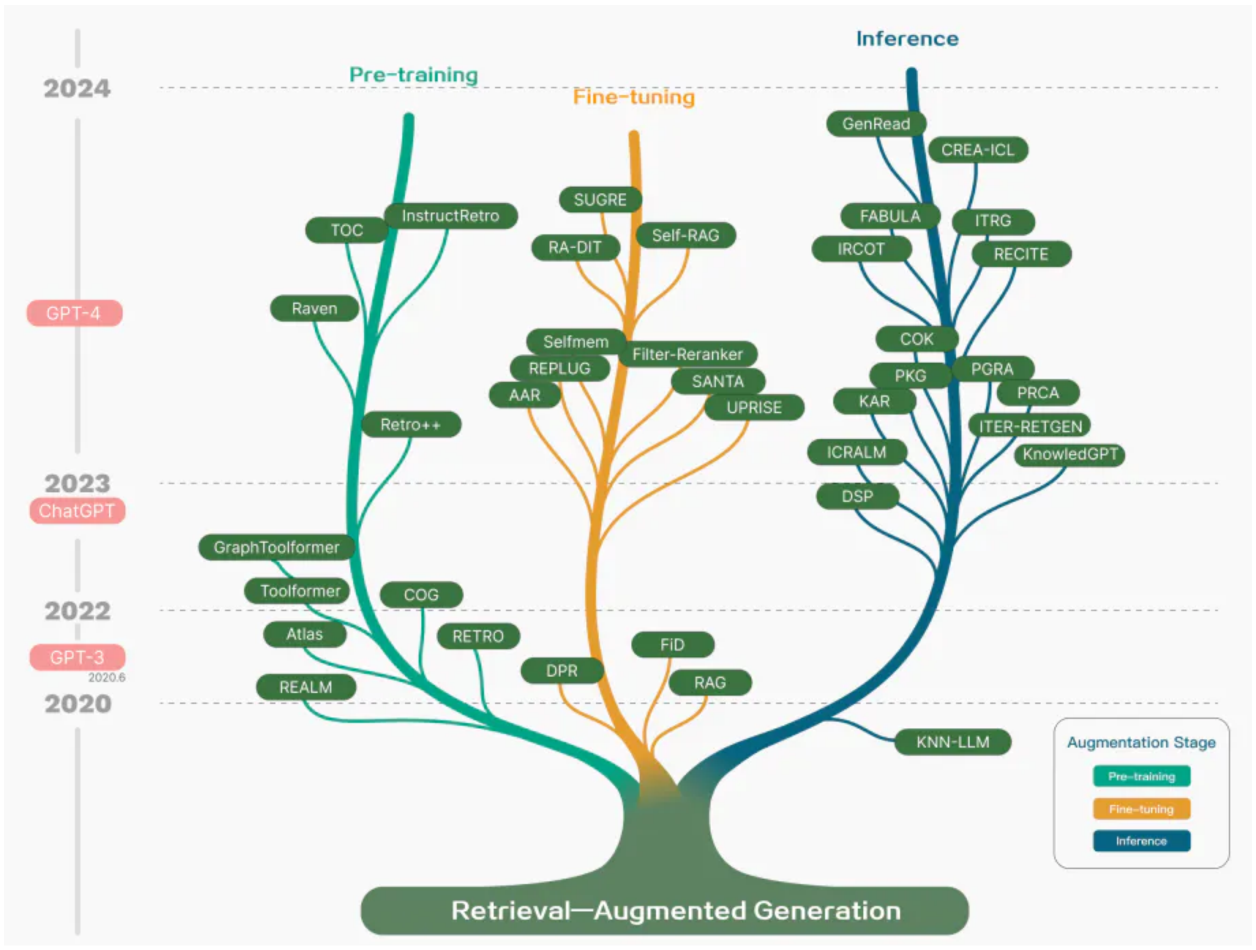


Figure 2: alt text

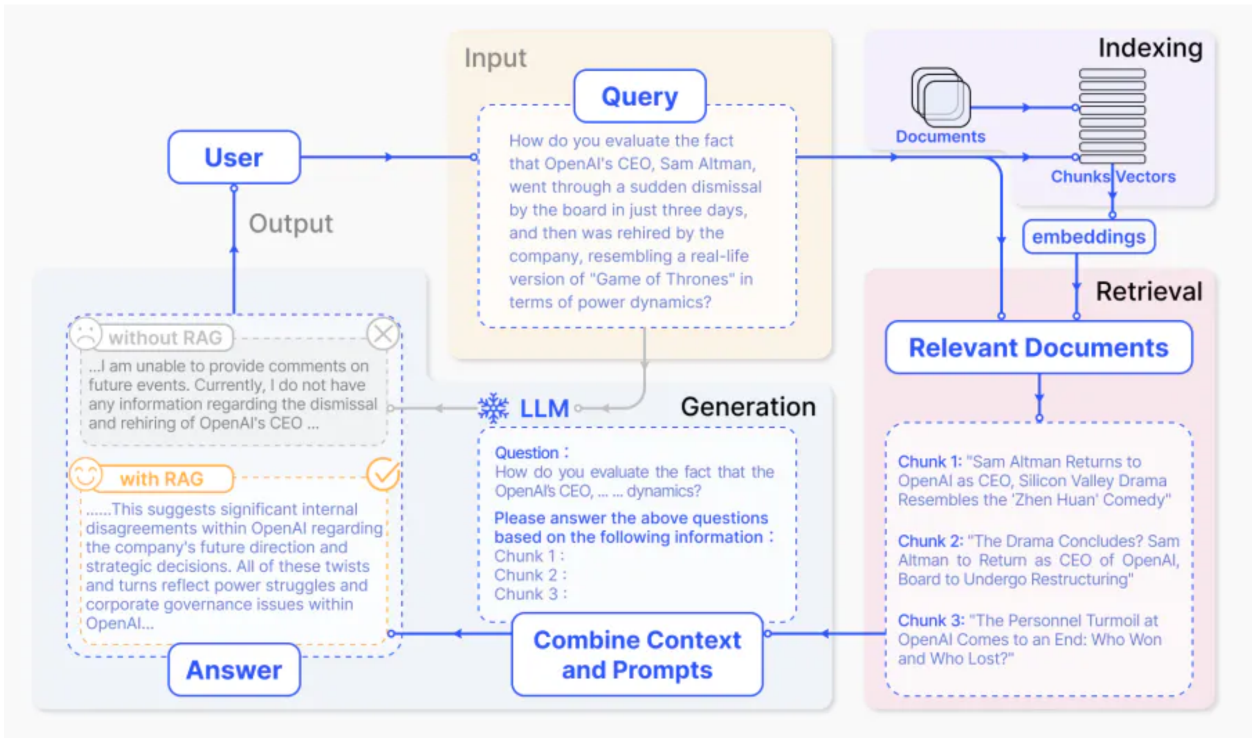

Input
Query
How do you evaluate the fact that OpenAI's CEO, Sam Altman, went through a sudden dismissal by the board in just three days, and then was rehired by the company, resembling a real-life version of "Game of Thrones" in terms of power dynamics?
User
Output
Indexing
Documents
Chunks
Vectors
embeddings
Retrieval
Relevant Documents
without RAG
...I am unable to provide comments on future events. Currently, I do not have any information regarding the dismissal and rehiring of OpenAI's CEO ...
with RAG
......This suggests significant internal disagreements within OpenAI regarding the company's future direction and strategic decisions. All of these twists and turns reflect power struggles and corporate governance issues within OpenAI...
Answer
LLM
Generation
Question :
How do you evaluate the fact that the OpenAI's CEO, ... dynamics?
Please answer the above questions based on the following information :
Chunk 1 :
Chunk 2 :
Chunk 3 :
Combine Context and Prompts
Chunk 1: "Sam Altman Returns to OpenAI as CEO, Silicon Valley Drama Resembles the 'Zhen Huan' Comedy"
Chunk 2: "The Drama Concludes? Sam Altman to Return as CEO of OpenAI, Board to Undergo Restructuring"
Chunk 3: "The Personnel Turmoil at OpenAI Comes to an End: Who Won and Who Lost?"

Figure 3: alt text

RAG Components

- **Input**: The question to which the LLM system responds. If no RAG is used, the LLM directly responds to the question.
- **Indexing**: If RAG is used, related documents are indexed by chunking them, generating embeddings of the chunks, and indexing them into a vector store. At inference, the query is also embedded similarly.
- **Retrieval**: Relevant documents are obtained by comparing the query against the indexed vectors, also denoted as “Relevant Documents”.
- **Generation**: Relevant documents are combined with the original prompt as additional context. The combined text and prompt are then passed to the model for response generation, which is then prepared as the final output of the system to the user.

RAG Paradigms

RAG systems have evolved from Naive RAG to Advanced RAG and Modular RAG to address certain limitations around performance, cost, and efficiency.

Naive RAG

- **Process**: A user input is used to query relevant documents which are then combined with a prompt and passed to the model to generate a final response. Conversational history can be integrated into the prompt if the application involves multi-turn dialogue interactions.
- **Limitations**: Low precision (misaligned retrieved chunks), low recall (failure to retrieve all relevant chunks), outdated information passed to the LLM, hallucination issues, poor and inaccurate responses, redundancy and repetition when augmentation is applied, and over-dependence on the augmented information.

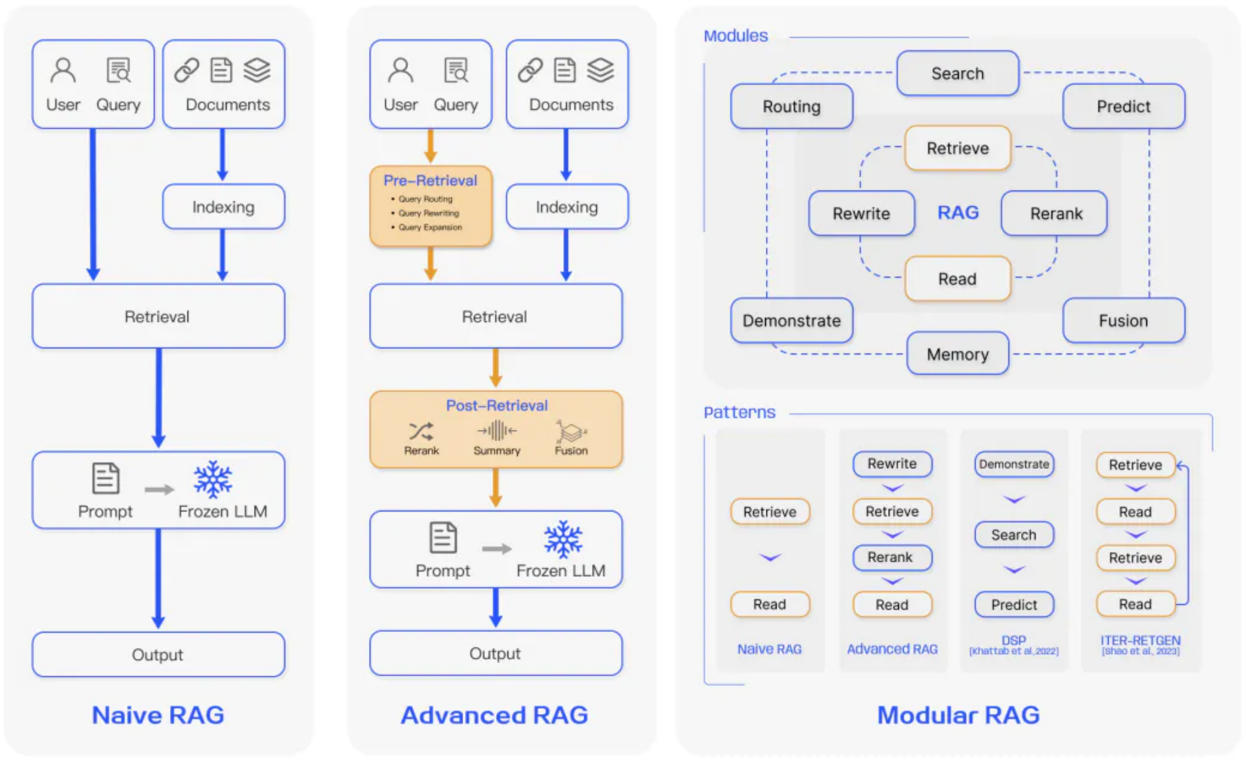


Figure 4: alt text

Advanced RAG

Advanced RAG helps deal with issues present in Naive RAG by improving retrieval quality through optimizing the pre-retrieval, retrieval, and post-retrieval processes.

- **Pre-retrieval Optimization**: Enhancing data granularity, optimizing index structures, adding metadata, alignment optimization, and mixed retrieval.
- **Retrieval Optimization**: Fine-tuning the embedding to optimize retrieval relevance or employing dynamic embeddings that better capture contextual understanding (e.g., OpenAI's embeddings-ada-02 model).
- **Post-retrieval Optimization**: Avoiding context window limits and dealing with noisy or potentially distracting information. Common approaches include re-ranking, relocation of relevant context to the edges of the prompt, recalculating the semantic similarity between the query and relevant text chunks, and prompt compression.

Modular RAG

- **Definition**: Enhances functional modules such as incorporating a search module for similarity retrieval and applying fine-tuning in the retriever. Both Naive RAG and Advanced RAG are special cases of Modular RAG and are made up of fixed modules.
- **Extended RAG Modules**: Search, memory, fusion, routing, predict, and task adapter which solve different problems. These modules can be rearranged to suit specific problem contexts.
- **Benefits**: Greater diversity and flexibility in that you can add or replace modules or adjust the flow between modules based on task requirements.

Optimization Techniques for RAG Pipelines

- **Hybrid Search Exploration**: Leverages a combination of search techniques like keyword-based search and semantic search to retrieve relevant and context-rich information.
- **Recursive Retrieval and Query Engine**: Involves a recursive retrieval process that might start with small semantic chunks and subsequently retrieve larger chunks that enrich the context.
- **StepBack-prompt**: A prompting technique that enables LLMs to perform abstraction that produces concepts and principles that guide reasoning.
- **Sub-Queries**: Different query strategies such as tree queries or sequential querying of chunks that can be used for different scenarios. LlamaIndex offers a sub question query engine that allows a query to be broken down into several questions that use different relevant data sources.
- **Hypothetical Document Embeddings (HyDE)**: Generates a hypothetical answer to a query, embeds it, and uses it to retrieve documents similar to the hypothetical answer as opposed to using the query directly.

RAG Framework

In this section, we summarize the key developments of the components of a RAG system, which include Retrieval, Generation, and Augmentation.

Retrieval

Retrieval is the component of RAG that deals with retrieving highly relevant context from a retriever. A retriever can be enhanced in many ways, including:

Enhancing Semantic Representations

- **Chunking**: Choosing the right chunking strategy depends on the content and the application. Different models perform better on varying block sizes.
- **Fine-tuned Embedding Models**: Fine-tuning the embedding model may be required if working with a specialized domain.

Aligning Queries and Documents

- **Query Rewriting**: Rewriting queries using a variety of techniques such as Query2Doc, ITER-RETGEN, and HyDE.
- **Embedding Transformation**: Optimizing the representation of query embeddings and aligning them to a latent space that is more closely aligned with a task.

Aligning Retriever and LLM

- **Fine-tuning Retrievers**: Using an LLM's feedback signals to refine the retrieval models.
- **Adapters**: Incorporating external adapters to help with the alignment process.

Generation

The generator in a RAG system is responsible for converting retrieved information into a coherent text that will form the final output of the model. This process involves diverse input data which sometimes require efforts to refine the adaptation of the language model to the input data derived from queries and documents. This can be addressed using post-retrieval process and fine-tuning:

- **Post-retrieval with Frozen LLM**: Post-retrieval processing leaves the LLM untouched and instead focuses on enhancing the quality of retrieval results through operations like information compression and result reranking.
- **Fine-tuning LLM for RAG**: To improve the RAG system, the generator can be further optimized or fine-tuned to ensure that the generated text is natural and effectively leverages the retrieved documents.

Augmentation

Augmentation involves the process of effectively integrating context from retrieved passages with the current generation task. Before discussing more on the augmentation process, augmentation stages, and augmentation data, here is a taxonomy of RAG's core components:

Retrieval Augmentation

- **Stages**: Can be applied in pre-training, fine-tuning, and inference stages.
- **RETRO**: An example of a system that leverages retrieval augmentation for large-scale pre-training from scratch.

Augmentation Stages

- **Pre-training**: RETRO uses an additional encoder built on top of external knowledge.
- **Fine-tuning**: Can be combined with RAG to help develop and improve the effectiveness of RAG systems.
- **Inference**: Many techniques are applied to effectively incorporate retrieved content to meet specific task demands and further refine the RAG process.

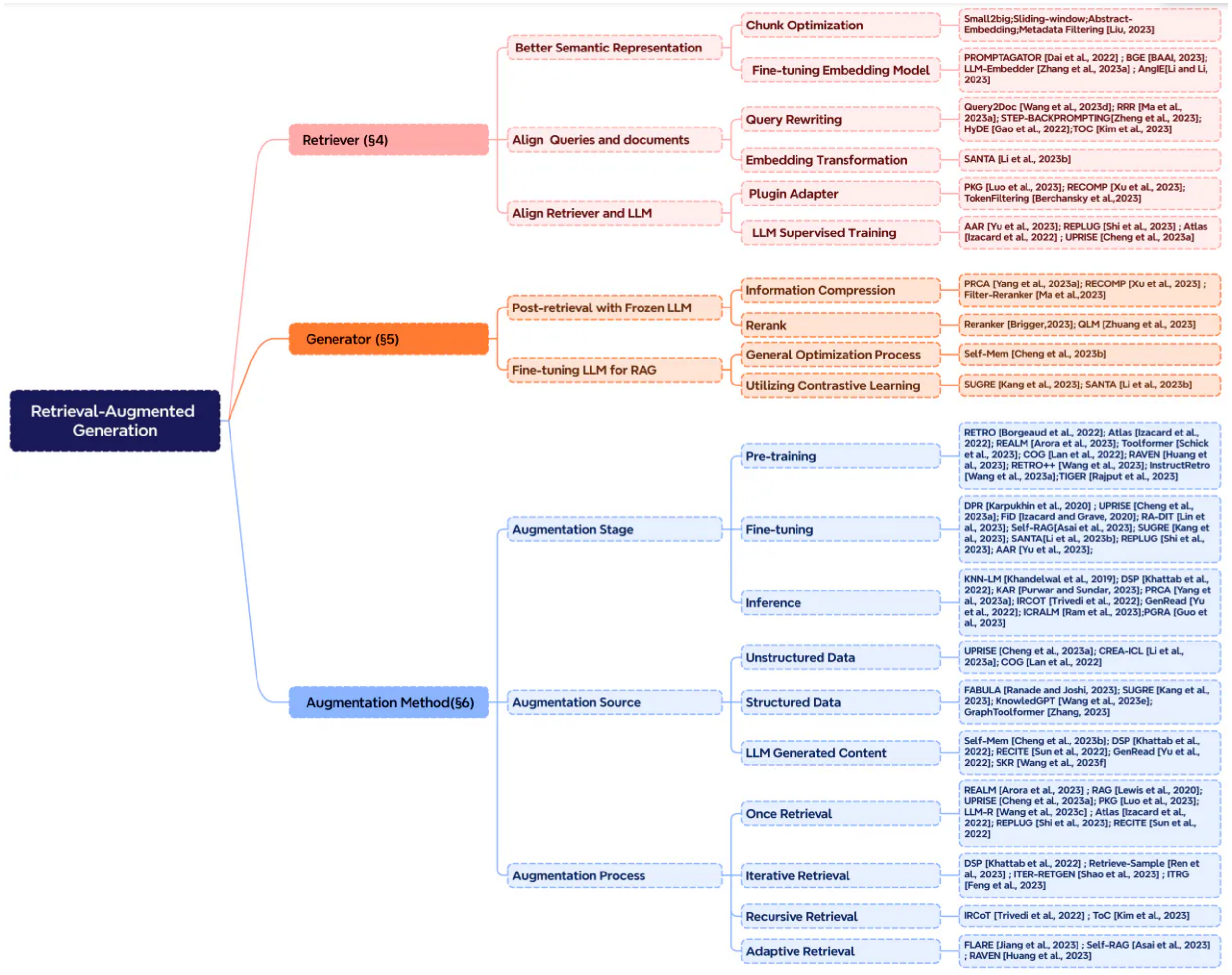

Retrieval-Augmented Generation
Retriever (§4)
Better Semantic Representation
Chunk Optimization
Small2big;Sliding-window;Abstract-Embedding;Metadata Filtering [Liu, 2023]
Fine-tuning Embedding Model
PROMPTAGATOR [Dai et al., 2022] ; BGE [BAAI, 2023]; LLM-Embedder [Zhang et al., 2023a] ; AnglE[Li and Li, 2023]
Align Queries and documents
Query Rewriting
Query2Doc [Wang et al., 2023d]; RRR [Ma et al., 2023a]; STEP-BACKPROMPTING[Zheng et al., 2023]; HyDE [Gao et al., 2022];TOC [Kim et al., 2023]
Embedding Transformation
SANTA [Li et al., 2023b]
Align Retriever and LLM
Plugin Adapter
PKG [Luo et al., 2023]; RECOMP [Xu et al., 2023]; TokenFiltering [Berchansky et al.,2023]
LLM Supervised Training
AAR [Yu et al., 2023]; REPLUG [Shi et al., 2023] ; Atlas [Izacard et al., 2022] ; UPRISE [Cheng et al., 2023a]
Generator (§5)
Post-retrieval with Frozen LLM
Information Compression
PRCA [Yang et al., 2023a]; RECOMP [Xu et al., 2023] ; Filter-Reranker [Ma et al.,2023]
Rerank
Reranker [Brigger,2023]; QLM [Zhuang et al., 2023]
Fine-tuning LLM for RAG
General Optimization Process
Self-Mem [Cheng et al., 2023b]
Utilizing Contrastive Learning
SUGRE [Kang et al., 2023]; SANTA [Li et al., 2023b]
Augmentation Method(§6)
Augmentation Stage
Pre-training
RETRO [Borgeaud et al., 2022]; Atlas [Izacard et al., 2022]; REALM [Arora et al., 2023]; Toolformer [Schick et al., 2023]; COG [Lan et al., 2022]; RAVEN [Huang et al., 2023]; RETRO++ [Wang et al., 2023]; InstructRetro [Wang et al., 2023a];TIGER [Rajput et al., 2023]
Fine-tuning
DPR [Karpukhin et al., 2020] ; UPRISE [Cheng et al., 2023a]; FiD [Izacard and Grave, 2020]; RA-DIT [Lin et al., 2023]; Self-RAG[Asai et al., 2023]; SUGRE [Kang et al., 2023]; SANTA[Li et al., 2023b]; REPLUG [Shi et al., 2023]; AAR [Yu et al., 2023];
Inference
KNN-LM [Khandelwal et al., 2019]; DSP [Khattab et al., 2022]; KAR [Purwar and Sundar, 2023]; PRCA [Yang et al., 2023a]; IRCOT [Trivedi et al., 2022]; GenRead [Yu et al., 2022]; ICRALM [Ram et al., 2023];PGRA [Guo et al., 2023]
Augmentation Source
Unstructured Data
UPRISE [Cheng et al., 2023a]; CREA-ICL [Li et al., 2023a]; COG [Lan et al., 2022]
Structured Data
FABULA [Ranade and Joshi, 2023]; SUGRE [Kang et al., 2023]; KnowledGPT [Wang et al., 2023e]; GraphToolformer [Zhang, 2023]
LLM Generated Content
Self-Mem [Cheng et al., 2023b]; DSP [Khattab et al., 2022]; RECITE [Sun et al., 2022]; GenRead [Yu et al., 2022]; SKR [Wang et al., 2023f]
Augmentation Process
Once Retrieval
REALM [Arora et al., 2023] ; RAG [Lewis et al., 2020]; UPRISE [Cheng et al., 2023a]; PKG [Luo et al., 2023]; LLM-R [Wang et al., 2023c] ; Atlas [Izacard et al., 2022]; REPLUG [Shi et al., 2023]; RECITE [Sun et al., 2022]
Iterative Retrieval
DSP [Khattab et al., 2022] ; Retrieve-Sample [Ren et al., 2023] ; ITER-RETGEN [Shao et al., 2023] ; ITRG [Feng et al., 2023]
Recursive Retrieval
IRCoT [Trivedi et al., 2022] ; ToC [Kim et al., 2023]
Adaptive Retrieval
FLARE [Jiang et al., 2023] ; Self-RAG [Asai et al., 2023] ; RAVEN [Huang et al., 2023]

Figure 5: alt text

Augmentation Source

- **Data Categories**: Unstructured, structured, and LLM-generated data.

Augmentation Process

- **Iterative Retrieval**: Enables the model to perform multiple retrieval cycles to enhance the depth and relevance of information. Notable approaches include RETRO and GAR-meets-RAG.
- **Recursive Retrieval**: Recursively iterates on the output of one retrieval step as the input to another retrieval step. Notable approaches include IRCoT and Tree of Clarifications.
- **Adaptive Retrieval**: Tailors the retrieval process to specific demands by determining optimal moments and content for retrieval. Notable approaches include FLARE and Self-RAG.

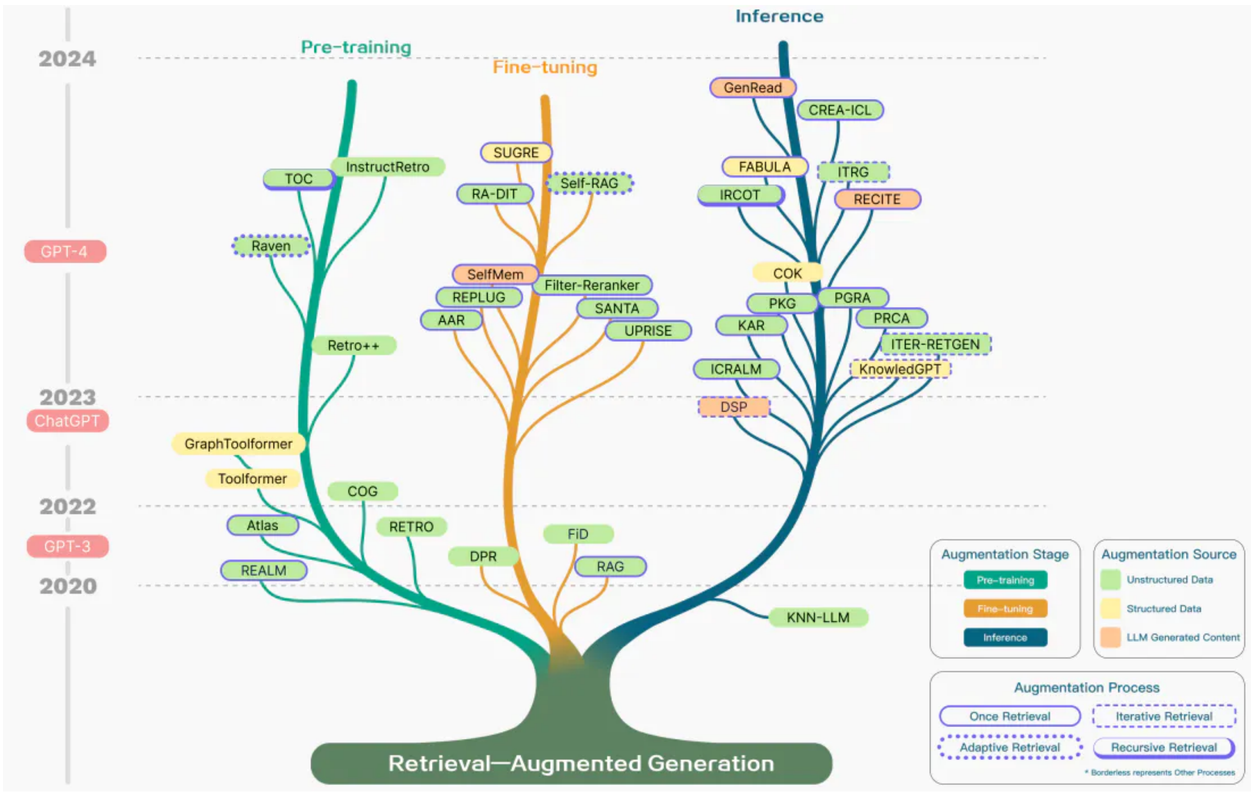


Figure 6: alt text

RAG vs. Fine-tuning

- **RAG**: Useful for integrating new knowledge.
- **Fine-tuning**: Can be used to improve model performance and efficiency through improving internal knowledge, output format, and teaching complex instruction following.
- **Prompting Engineering**: Can also help to optimize results by leveraging the inherent capabilities of the model.

RAG Evaluation

- **Role**: Plays a key role in understanding and optimizing the performance of RAG models across diverse application scenarios.
- **Targets**: Determined for both retrieval and generation where the goal is to evaluate both the quality of the context retrieved and the quality of the content generated.

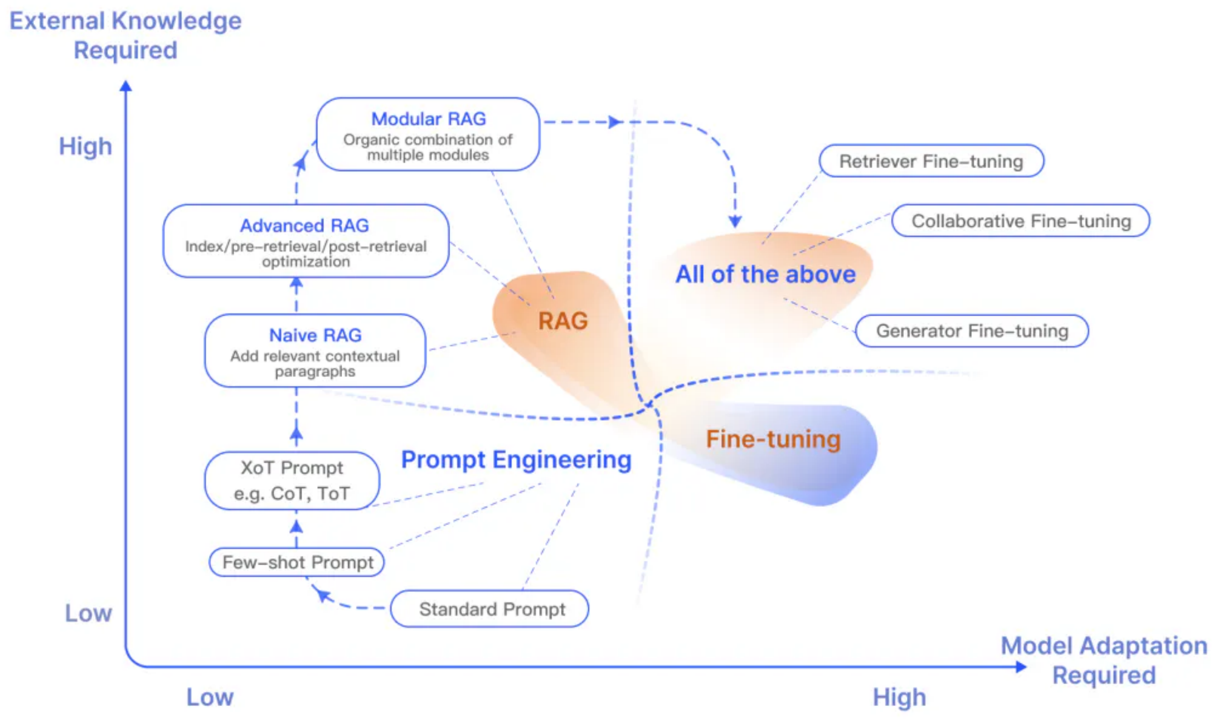

External Knowledge Required
High
Low
Modular RAG
Organic combination of multiple modules
Advanced RAG
Index/pre-retrieval/post-retrieval optimization
Naive RAG
Add relevant contextual paragraphs
XoT Prompt
e.g. CoT, ToT
Few-shot Prompt
Standard Prompt
Prompt Engineering
RAG
All of the above
Retriever Fine-tuning
Collaborative Fine-tuning
Generator Fine-tuning
Fine-tuning
Model Adaptation Required
Low
High

Figure 7: alt text

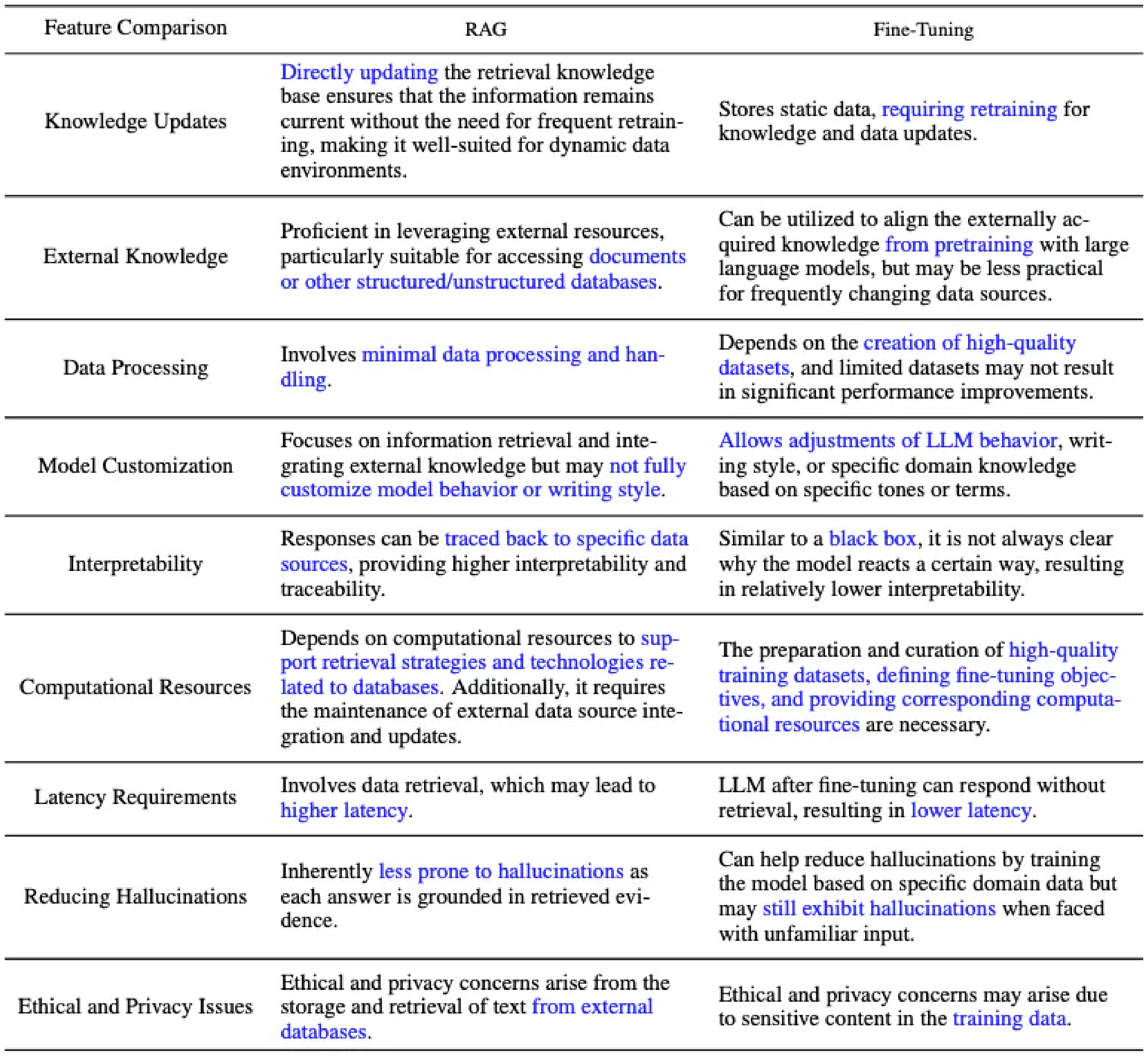
- **Retrieval Quality Metrics**: Metrics used in other knowledge-intensive domains like recommendation systems and information retrieval are used such as NDCG and Hit Rate.
- **Generation Quality Metrics**: Different aspects like relevance and harmfulness if it's unlabeled content or accuracy for labeled content.
- **Evaluation Methods**: Can involve either manual or automatic evaluation methods.

Evaluating a RAG Framework

- **Quality Scores**: Measuring context relevance, answer faithfulness, and answer relevance.
- **Abilities**: Noise robustness, negative rejection, information integration, and counterfactual robustness.

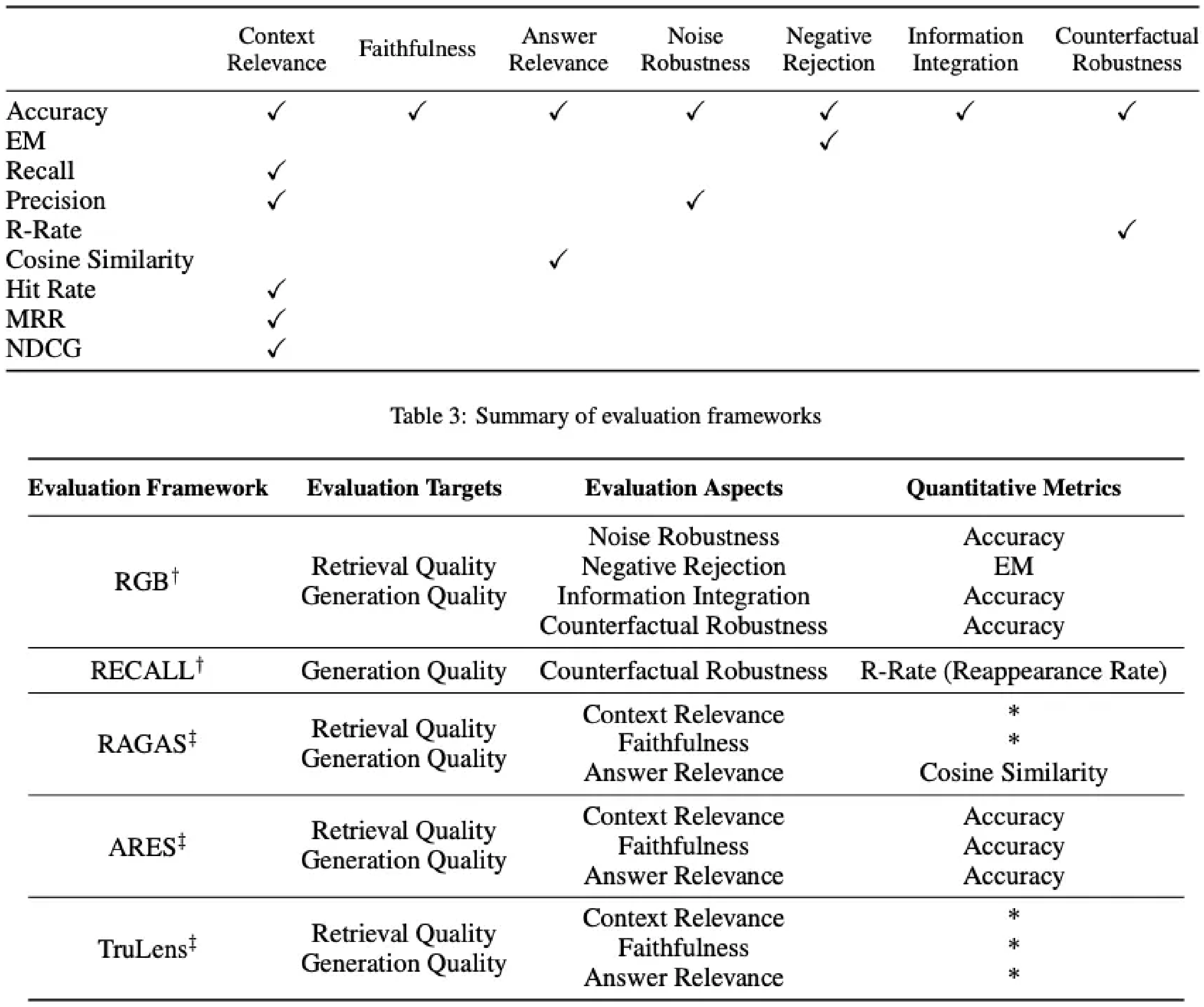
Challenges & Future of RAG

- **Context Length**: LLMs continue to extend context window size which presents challenges to how RAG needs to be adapted to ensure highly relevant and important context is captured.
- **Robustness**: Dealing with counterfactual and adversarial information is important to measure and improve in RAG.
- **Hybrid Approaches**: There is an ongoing research effort to better understand how to best optimize the use of both RAG and fine-tuned models.
- **Expanding LLM Roles**: Increasing the role and capabilities of LLMs to further enhance RAG systems is of high interest.
- **Scaling Laws**: Investigation of LLM scaling laws and how they apply to RAG systems are still not properly understood.
- **Production-ready RAG**: Production-grade RAG systems demand engineering excellence across performance, efficiency, data security, privacy, and more.
- **Multimodal RAG**: There is increasing interest in extending modalities for a RAG system to support tackling problems in more domains such as image, audio and video, code, and more.



Feature Comparison	RAG	Fine-Tuning
Knowledge Updates	Directly updating the retrieval knowledge base ensures that the information remains current without the need for frequent retraining, making it well-suited for dynamic data environments.	Stores static data, requiring retraining for knowledge and data updates.
External Knowledge	Proficient in leveraging external resources, particularly suitable for accessing documents or other structured/unstructured databases.	Can be utilized to align the externally acquired knowledge from pretraining with large language models, but may be less practical for frequently changing data sources.
Data Processing	Involves minimal data processing and handling.	Depends on the creation of high-quality datasets, and limited datasets may not result in significant performance improvements.
Model Customization	Focuses on information retrieval and integrating external knowledge but may not fully customize model behavior or writing style.	Allows adjustments of LLM behavior, writing style, or specific domain knowledge based on specific tones or terms.
Interpretability	Responses can be traced back to specific data sources, providing higher interpretability and traceability.	Similar to a black box, it is not always clear why the model reacts a certain way, resulting in relatively lower interpretability.
Computational Resources	Depends on computational resources to support retrieval strategies and technologies related to databases. Additionally, it requires the maintenance of external data source integration and updates.	The preparation and curation of high-quality training datasets, defining fine-tuning objectives, and providing corresponding computational resources are necessary.
Latency Requirements	Involves data retrieval, which may lead to higher latency.	LLM after fine-tuning can respond without retrieval, resulting in lower latency.
Reducing Hallucinations	Inherently less prone to hallucinations as each answer is grounded in retrieved evidence.	Can help reduce hallucinations by training the model based on specific domain data but may still exhibit hallucinations when faced with unfamiliar input.
Ethical and Privacy Issues	Ethical and privacy concerns arise from the storage and retrieval of text from external databases.	Ethical and privacy concerns may arise due to sensitive content in the training data.

Figure 8: alt text



	Context Relevance	Faithfulness	Answer Relevance	Noise Robustness	Negative Rejection	Information Integration	Counterfactual Robustness
Accuracy	✓	✓	✓	✓	✓	✓	✓
EM					✓		
Recall	✓						
Precision	✓			✓			
R-Rate							✓
Cosine Similarity			✓				
Hit Rate	✓						
MRR	✓						
NDCG	✓						

Table 3: Summary of evaluation frameworks

Evaluation Framework	**Evaluation Targets**	**Evaluation Aspects**	**Quantitative Metrics**
RGB†	Retrieval Quality Generation Quality	Noise Robustness Negative Rejection Information Integration Counterfactual Robustness	Accuracy EM Accuracy Accuracy
RECALL†	Generation Quality	Counterfactual Robustness	R-Rate (Reappearance Rate)
RAGAS‡	Retrieval Quality Generation Quality	Context Relevance Faithfulness Answer Relevance	* * Cosine Similarity
ARES‡	Retrieval Quality Generation Quality	Context Relevance Faithfulness Answer Relevance	Accuracy Accuracy Accuracy
TruLens‡	Retrieval Quality Generation Quality	Context Relevance Faithfulness Answer Relevance	* * *

Figure 9: alt text

- **Evaluation**: The interest in building complex applications with RAG requires special attention to develop nuanced metrics and assessment tools that can more reliably assess different aspects such as contextual relevance, creativity, content diversity, factuality, and more. In addition, there is also a need for better interpretability research and tools for RAG.