

# Intro of Retrieval Augmented Generation (RAG) and application demos

by Henry Heng LUO

# Content

## ❖ RAG Summary

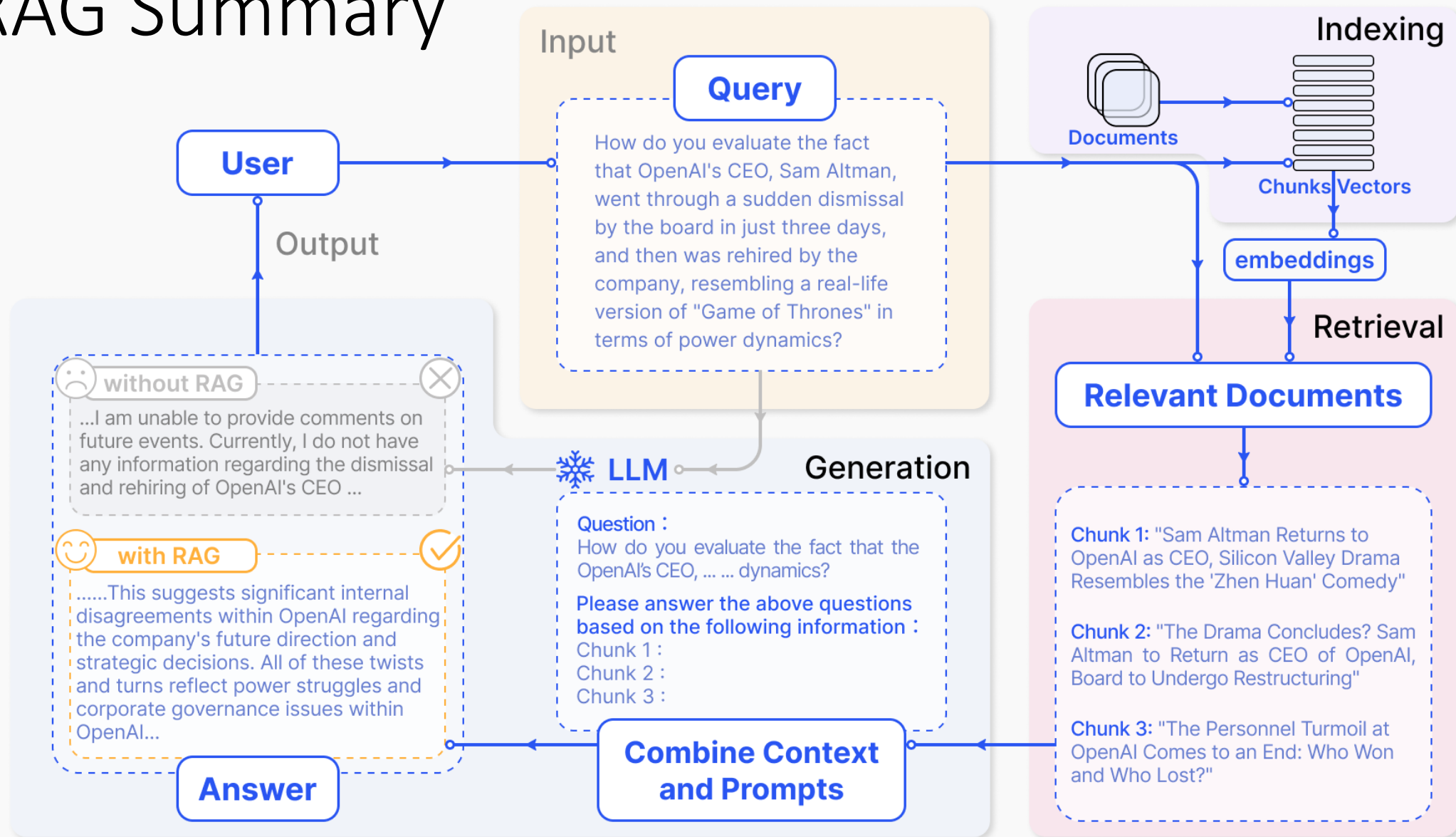
## ❖ Hands-on practices

1. PRACTICE of Basic RAG pipeline
2. PRACTICE of Sentence-window retrieval pipeline
3. PRACTICE of Auto-merging retrieval pipeline

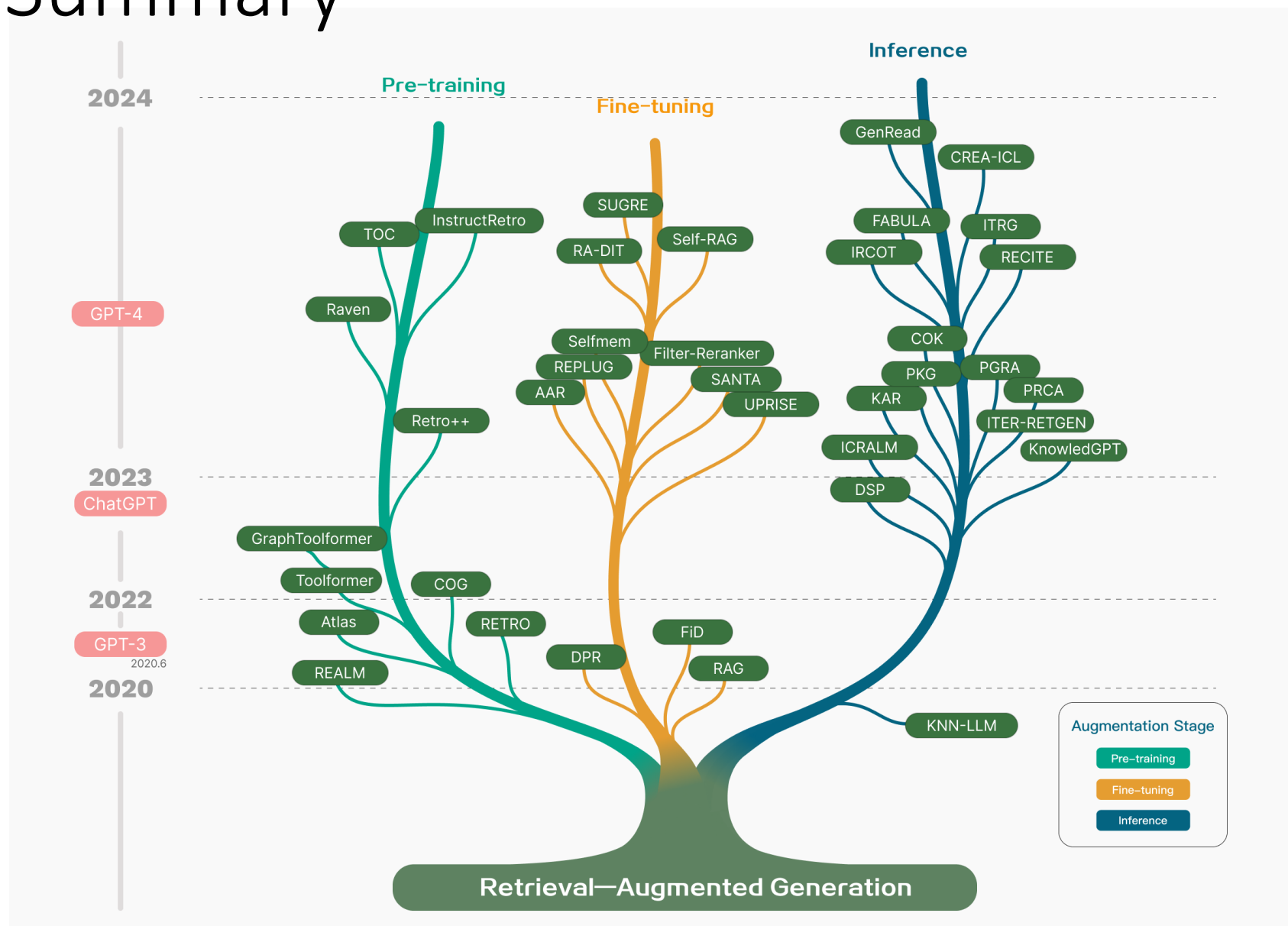
# RAG Summary

- Large Language Models are with intrinsic flaws.
- They are can produce misleading "**hallucinations**"
- They rely on potentially **outdated information**
- They are inefficient when dealing with **specific knowledge**
- They lack depth in **specialized fields**
- They fall short in **reasoning** abilities
- They lack **controllability**
- They cannot **trace** the knowledge source
- They cannot protect **data privacy**
- They are with **high cost** to train
- **Retrieval-Augmented Generation (RAG)** significantly improves the precision and pertinence of content by first **retrieve relevant information** from an **external database** of documents prior to the language model's answer generation.

# RAG Summary



# RAG Summary



# RAG Summary

- **Basic RAG**

- The classic basic RAG process, also known as Naive RAG, mainly includes three basic steps:
  1. **Indexing** - Splitting the document corpus into shorter chunks and building a vector index through an encoder.
  2. **Retrieval** - Retrieving relevant document fragments based on the similarity between the question and the chunks.
  3. **Generation** - Generating an answer to the question conditioned on the retrieved context.

# RAG Summary

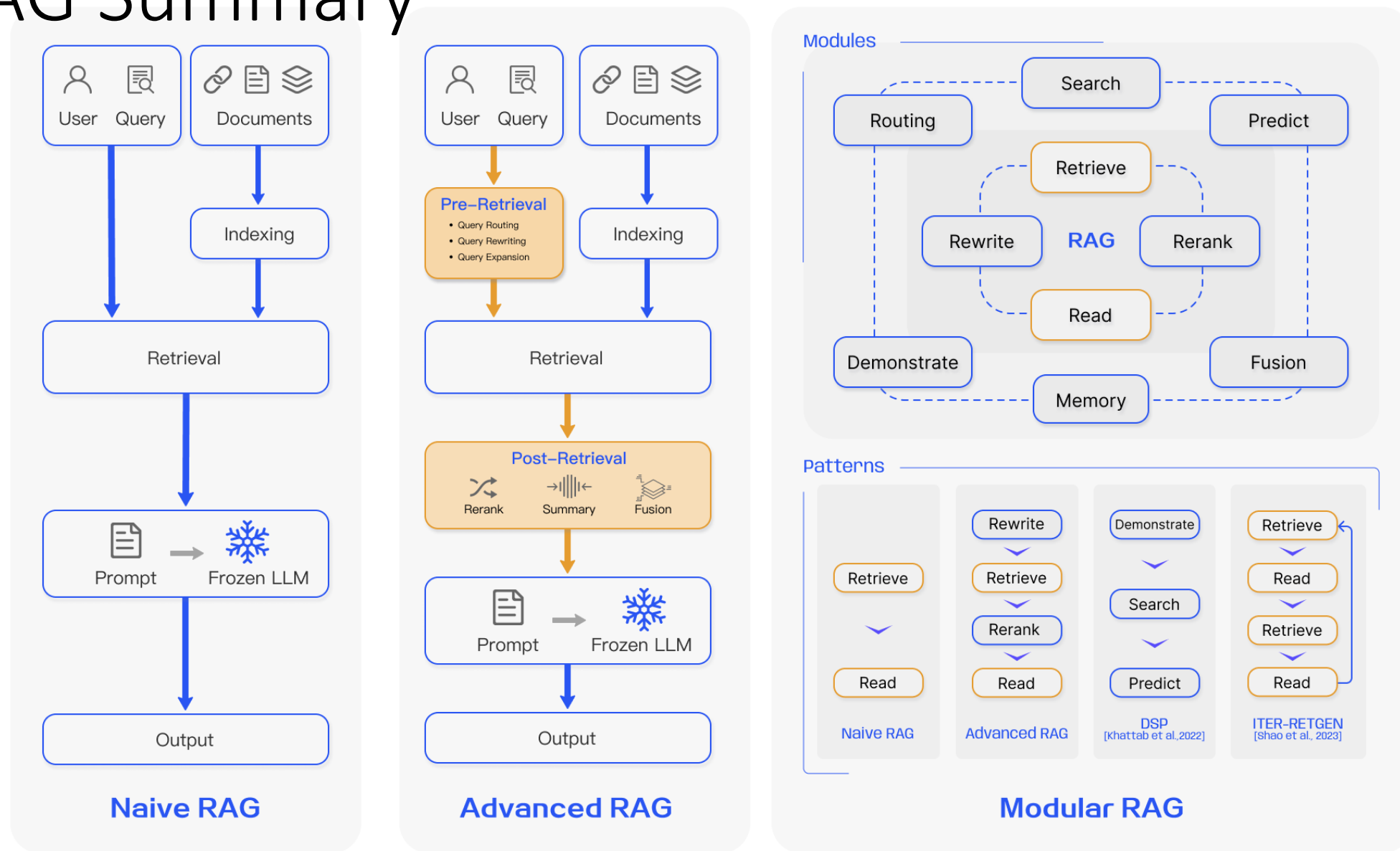
- **Advanced RAG**
- The Advanced RAG paradigm involves additional processing in **Pre-Retrieval** and **Post-Retrieval**.
  1. Before retrieval, methods such as **query rewriting, routing, and expansion** can be used to align the semantic differences between questions and document chunks.
  2. After retrieval, **rerank** the retrieved document corpus can avoid the "Lost in the Middle" phenomenon, or the context can be filtered and **compressed** to shorten the window length.

# RAG Summary

- **Modular RAG**
- Structurally, it is more free and flexible, introducing more specific functional modules, such as query search engines and the fusion of multiple answers.
- Technologically, it integrates retrieval with fine-tuning, reinforcement learning, and other techniques.
- In terms of process, the RAG modules are designed and orchestrated, resulting in various RAG patterns.



# RAG Summary



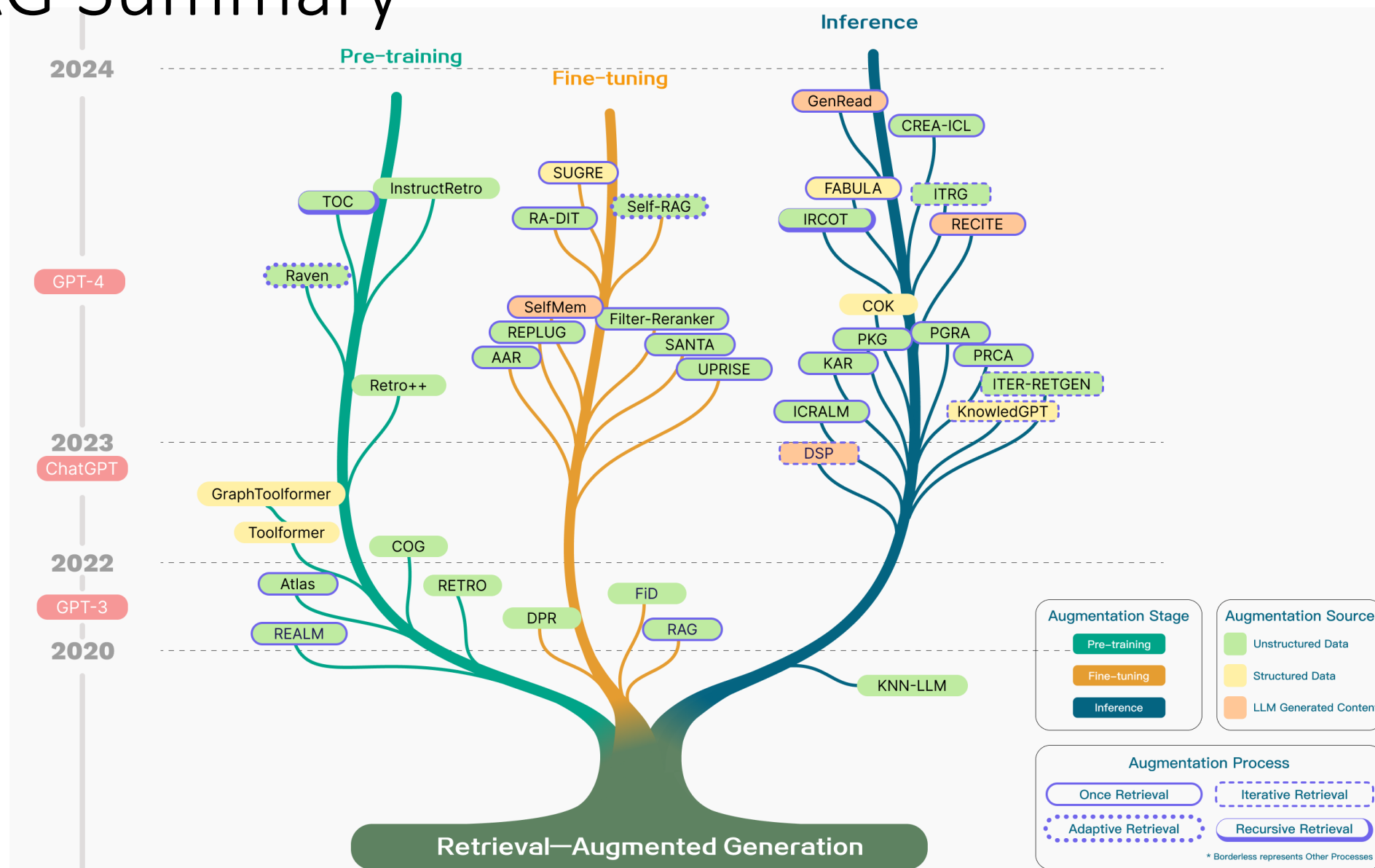
# RAG Summary

- To build a good RAG system, three critical questions need to be considered:
- **What** to retrieve?
- **When** to retrieve?
- **How** to use the retrieved content?

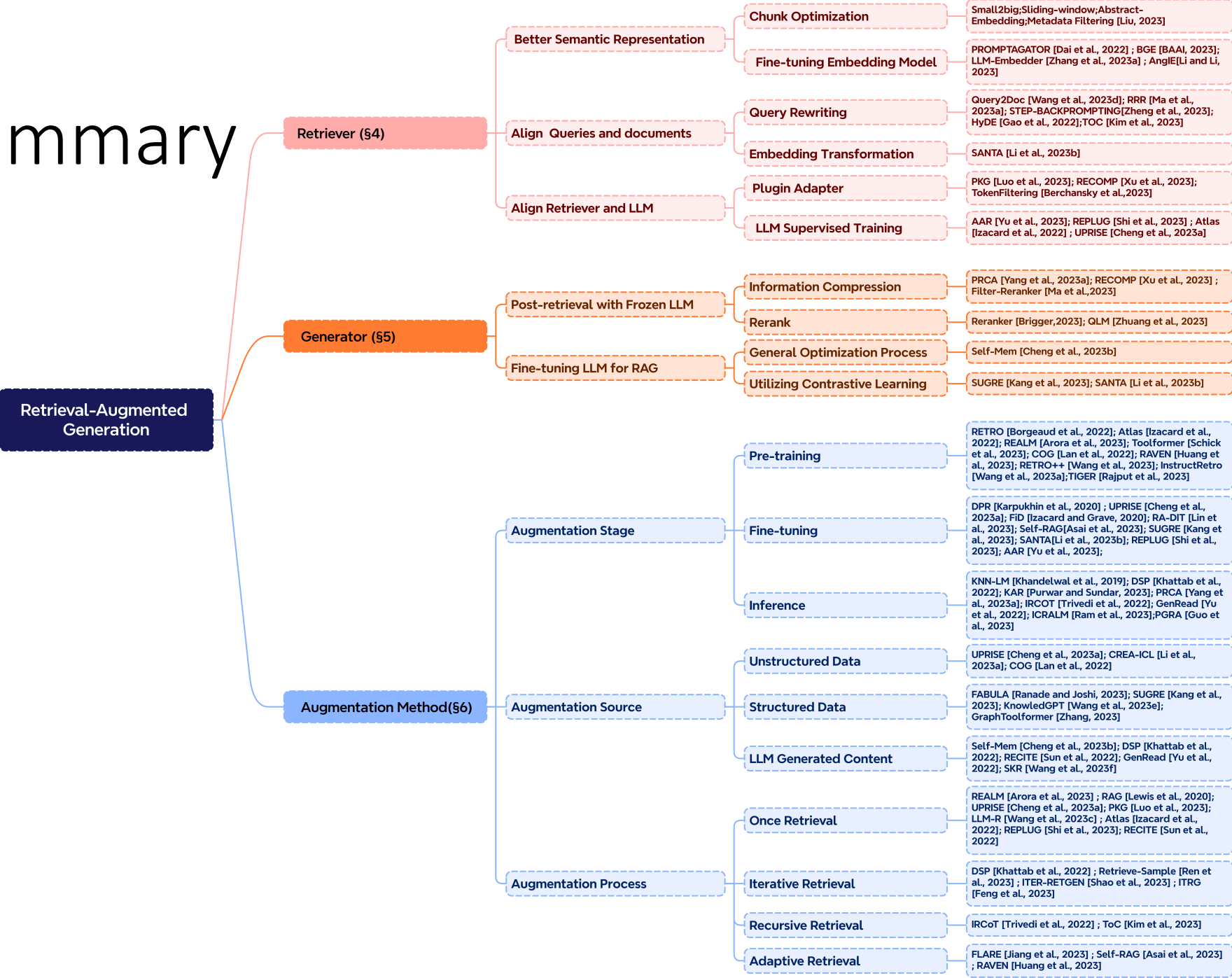
# RAG Summary

- **Augmentation Sources.** including **unstructured data** such as text paragraphs, phrases, or individual words. **Structured data** can also be used, such as indexed documents, triple data, or subgraphs, or retrieving from **content generated by LLMs** themselves.
- **Augmentation Stages.** performing during the **pre-training, fine-tuning, and inference** stages.
- **Augmentation process.** The initial retrieval was a **once** process, but **iterative** retrieval, **recursive** retrieval, and **adaptive** retrieval methods, where LLMs decide the timing of retrieval on their own, gradually emerged in the development of RAG.

# RAG Summary



# RAG Summary



# RAG Summary

- **RAG** is like giving the model a textbook for customized information retrieval, which is very suitable for specific queries.
- **Fine-tuning** is like a student internalizing knowledge over time, better suited for mimicking specific structures, styles, or formats.
- Depending on their reliance on external knowledge and requirements for model adjustment, they each have suitable scenarios.
- **To use RAG, Fine-tuning, Prompt Engineering together may yield the best results.**

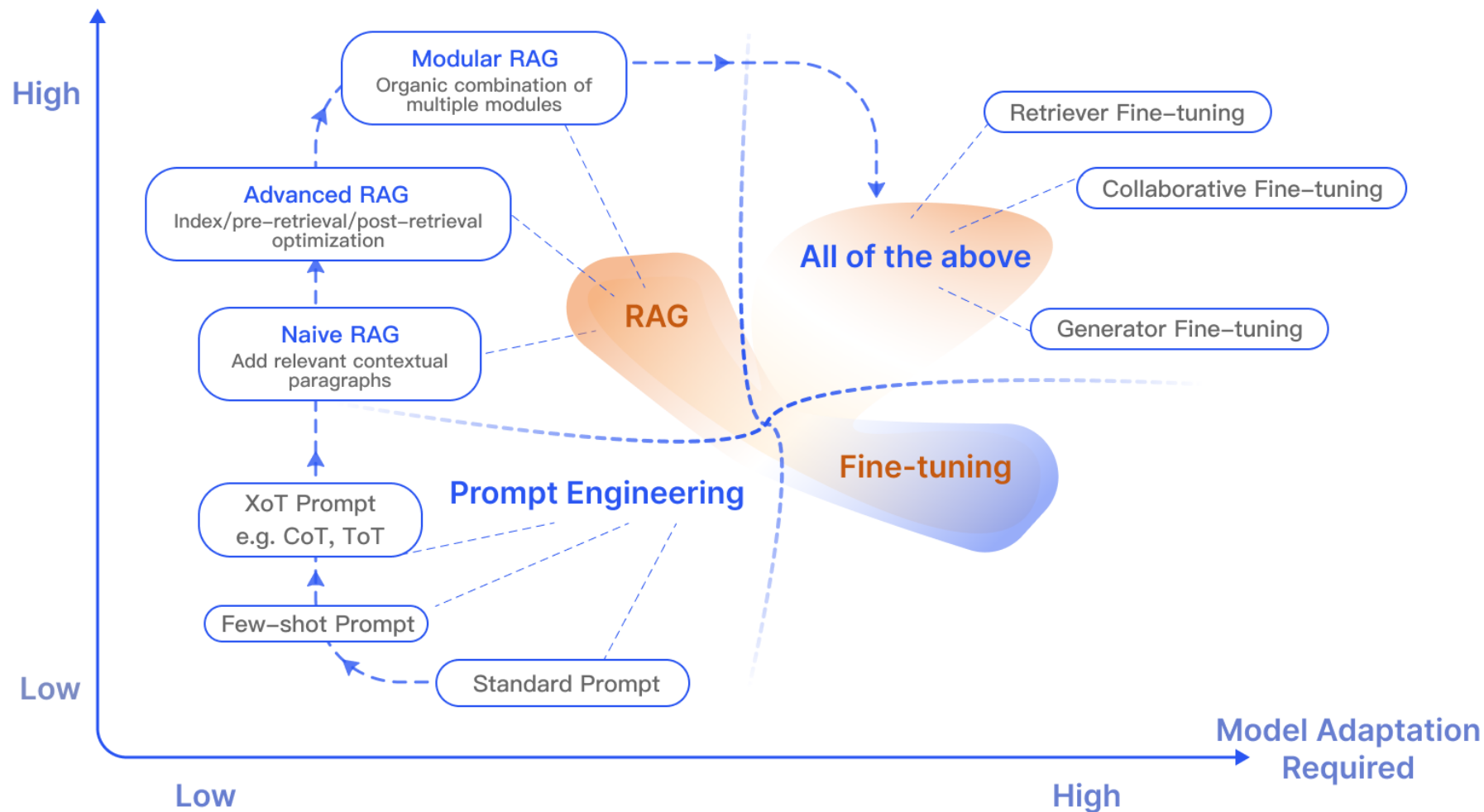
# RAG Summary

Table 1: Comparison between RAG and Fine-Tuning

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

# RAG Summary

External Knowledge  
Required





# RAG Summary

- The evaluation methods for RAG are diverse, mainly including three quality scores: **context relevance**, **answer fidelity**, and **answer relevance**.
- The evaluation involves four key capabilities: noise robustness, refusal ability, information integration, and counterfactual robustness.
- In terms of evaluation frameworks, there are benchmarks such as RGB and RECALL, as well as automated evaluation tools like RAGAS, ARES, and TruLens, which help to comprehensively measure the performance of RAG models.

# RAG Summary

Table 2: Summary of metrics applicable for evaluation aspects of RAG

	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	✓						

# RAG Summary

Table 3: Summary of evaluation frameworks

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

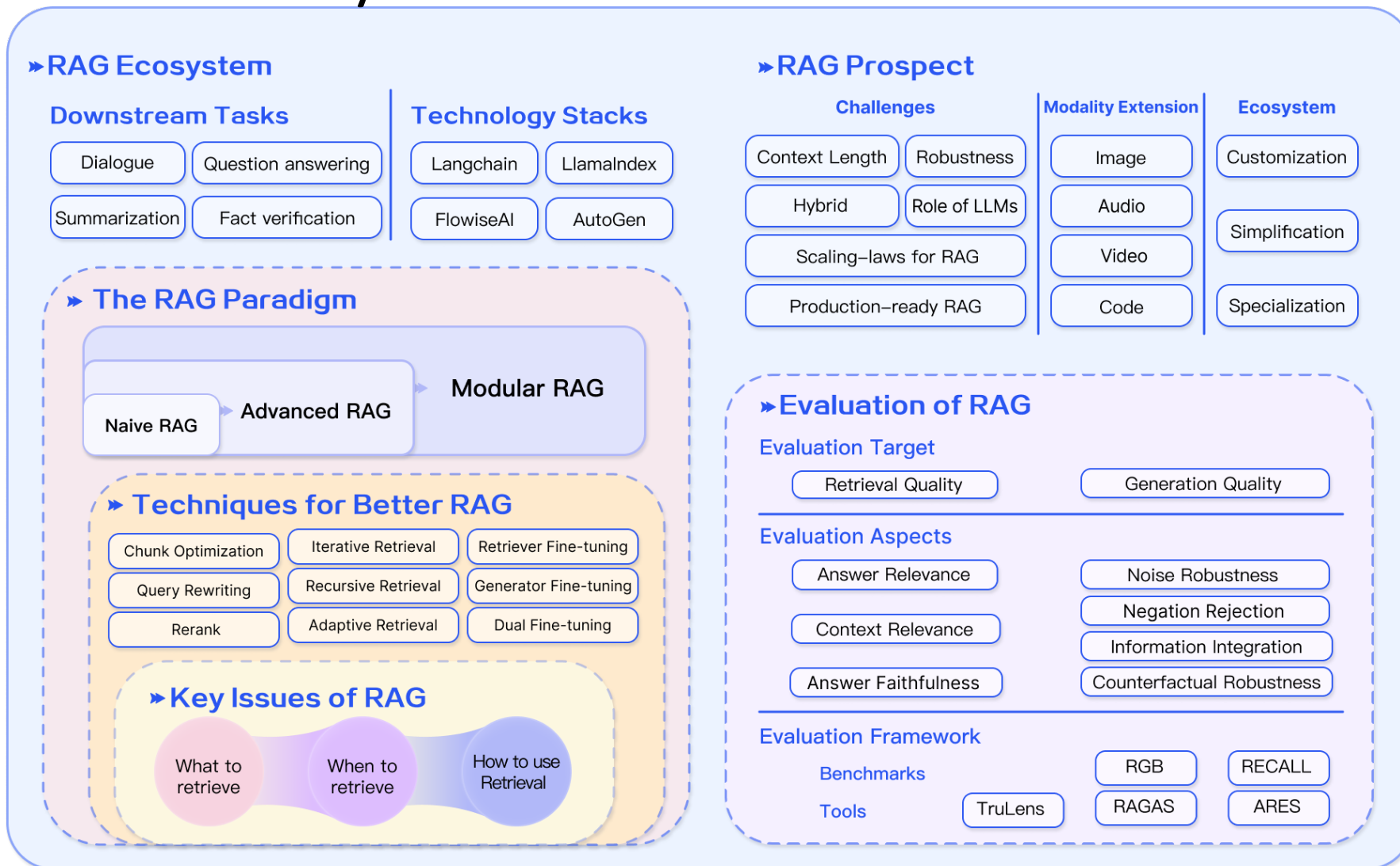
# RAG Summary

- To address the current challenges faced by RAG:
- **Context length.** What to do when the retrieved content is too much and exceeds the window limit? If the context window of LLMs is no longer limited, how should RAG be improved?
- **Robustness.** How to deal with incorrect content retrieved? How to filter and validate the retrieved content? How to enhance the model's resistance to poisoning and noise?
- **Coordination with fine-tuning.** How to leverage the effects of both RAG and FT simultaneously, how should they coordinate, organize, whether in series, alternation, or end-to-end?
- **Scaling Laws:** Does the RAG model satisfy the Scaling Law? Will RAG, or under what scenarios might RAG experience the phenomenon of Inverse Scaling Law?
- **The role of LLMs.** LLMs can be used for retrieval (replacing search with LLMs' generation or searching LLMs' memory), for generation, for evaluation. How to further explore the potential of LLMs in RAG?
- **Production-ready.** How to reduce the retrieval latency of ultra-large-scale corpora? How to ensure that the content retrieved is not leaked by LLMs
- **Multimodal Expansion.** How can the evolving technologies and concepts of RAG be extended to other modalities of data such as **images, audio, video, or code**?

# RAG Summary

- RAG can be applied to **question-answering systems** and more: such as **recommendation systems, information extraction, and report generation**.
- The RAG technology stack is booming. In addition to well-known tools like **Langchain** and **LlamaIndex**, the market is seeing an emergence of more targeted RAG tools, such as **customized tools and simplified tools**.

# RAG Summary

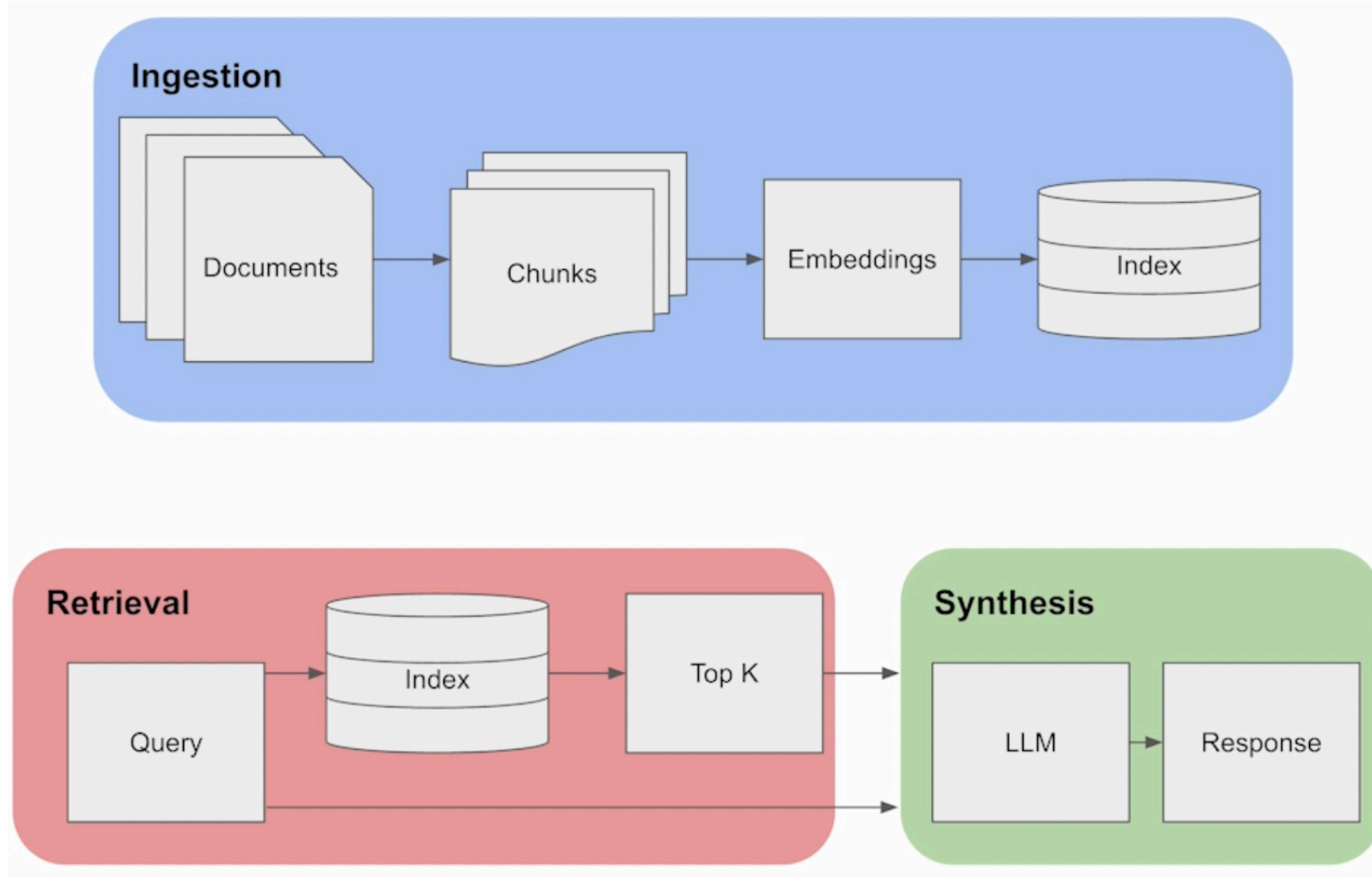


# PRACTICE of Basic RAG pipeline

- We want to infuse existing database information into the LLM.
- Each query will firstly send to retrieve the context information related to the existing database, (here vector database can be used), then the context information is wrapped in the prompt and sent to the LLM.
- Separate the documents into small chunk.
- Search the semantic matched small chunk.
- Return the top-k small chunks.

# PRACTICE of Basic RAG pipeline

The same text  
chunks are  
used in  
embeddings  
and synthesis

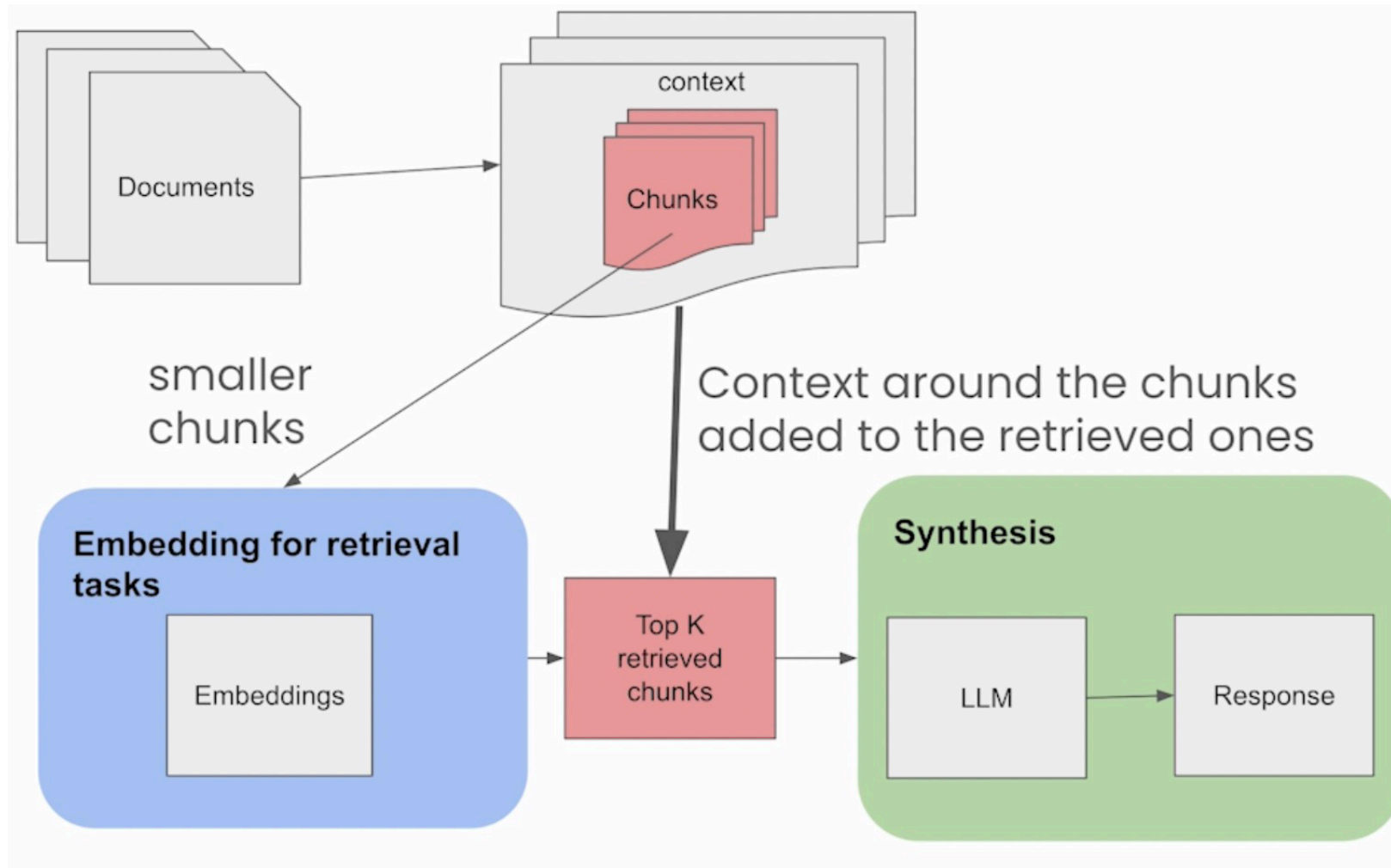




# PRACTICE of Sentence-window retrieval pipeline

- This is suitable for plenty context information, instead of only small chunk information.
- Separate the documents into sentence level.
- Search the semantic matched sentence chunk.
- Retrieve the sentence chunk with the previous and following sentences window, to form the context chunk.
- Rerank the context chunks.

# PRACTICE of Sentence-window retrieval pipeline



# PRACTICE of Sentence-window retrieval pipeline

Query: What  
are the  
concern  
surrounding  
the AMOC?

Continuous observation of the Atlantic meridional overturning circulation (AMOC) has improved the understanding of its variability (Frajka-Williams et al., 2019), but there is low confidence in the qualification of AMOC changes in the 20th century because of low agreement in quantitative reconstructed and simulated trends. Direct observational records since the mid-2000s remain too short to determine the relative contributions of internal variability, natural forcing and anthropogenic to AMOC change (high confidence). Over the 21st century, AMOC will very likely decline for all SSP scenarios but will not involve an abrupt collapse before 2100. 3.2.2.4 Sea Ice Changes Sea ice is a key driver of polar marine life, hosting unique ecosystems and affecting diverse marine organisms and food webs through its impact on light penetrations and supplies of nutrients and organic matter (Arrigo, 2014).

What the LLM sees

**Embedding Lookup**

What the LLM sees

# PRACTICE of Auto-merging retrieval pipeline

- The small chunk is good to match precisely, but we also need plenty context information.
- Define a hierarchy of smaller chunks.
- linked to parent chunks. If the set of smaller chunks linking to a parent chunk exceeds some threshold, then "merge" smaller chunks into the bigger parent chunk.
- Rerank the final parent chunks.

# PRACTICE of Auto-merging retrieval pipeline

Parent chunk

Chunk  
(512)

Chunk  
(128)

Chunk  
(128)

Chunk  
(128)

Chunk  
(128)

Auto-merging

Parent chunk

Chunk  
(512)

Chunk  
(128)

Chunk  
(128)

Chunk  
(128)

Chunk  
(128)

returned chunk

