Retrieval-Augmented Generation (RAG): Paradigms, Technologies, and Trends

Haofen Wang
Tongji University

CONTENTS

PART 01

Overview of RAG

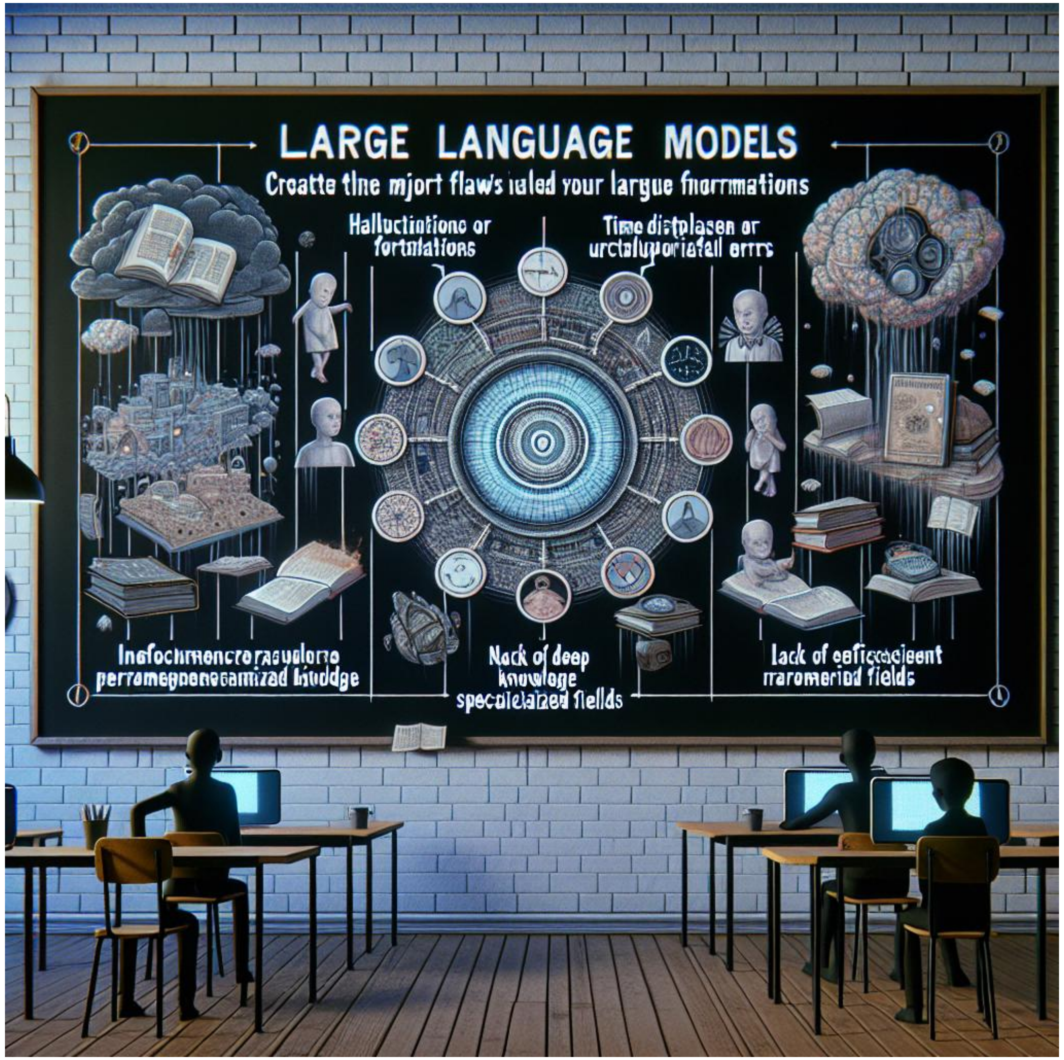
Background

Drawbacks of LLMs

- Hallucination
- Outdated information
- Low efficiency in parameterizing knowledge
- Lack of in-depth knowledge in specialized domains
- Weak inferential capabilities

Practical Requirements of Application

- Domain-specific accurate answering
- Frequent updates of data
- Traceability and explainability of generated content
- Controllable Cost
- Privacy protection of data



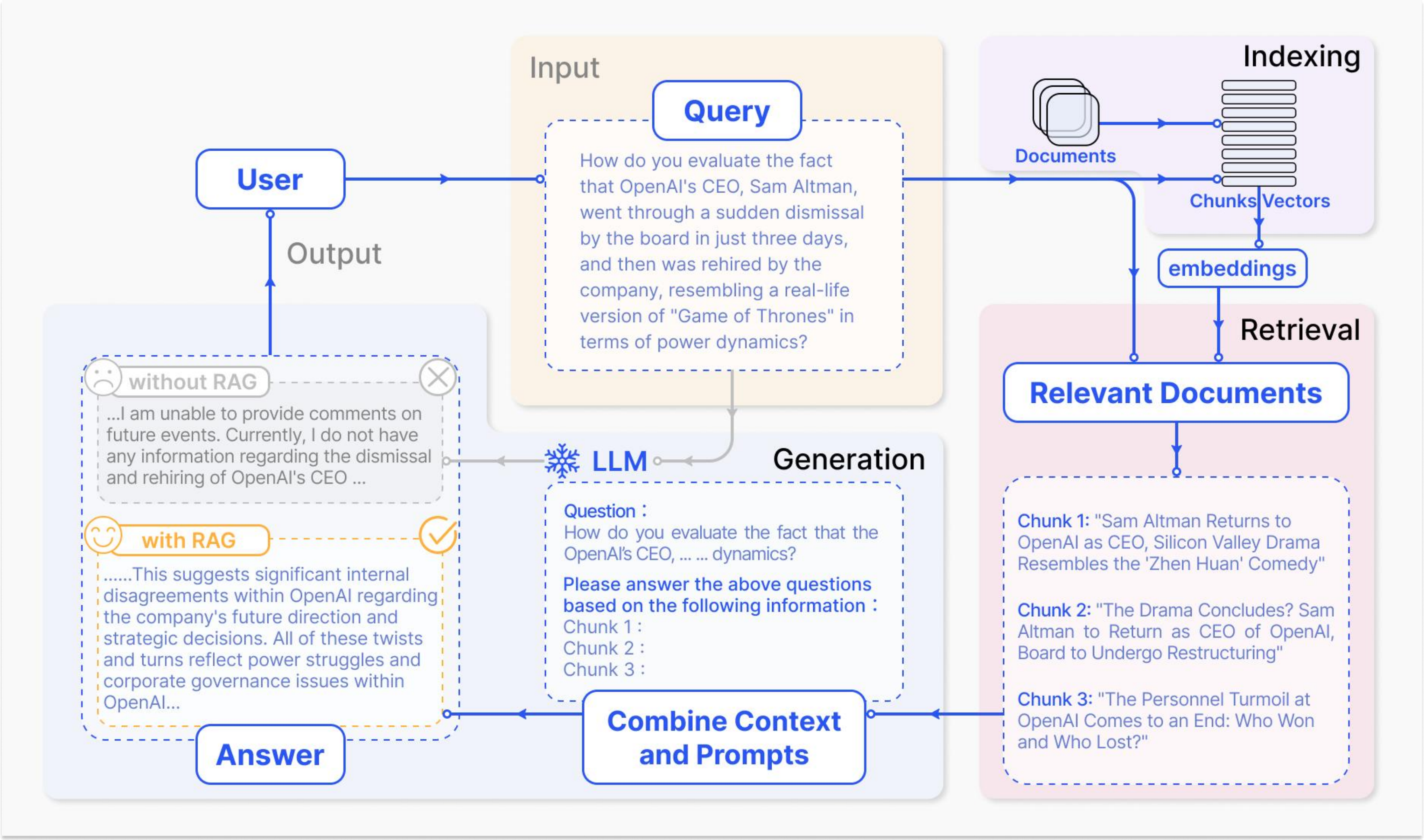
Draw by DALL·E-3

Retrieval-Augmented Generation (RAG)

When answering questions or generating text, it first retrieves relevant information from a large number of documents, and then LLMs generates answers based on this information.

By attaching a external knowledge base, there is no need to retrain the entire large model for each specific task.

The RAG model is especially suitable for knowledge-intensive tasks.



A typical case of RAG

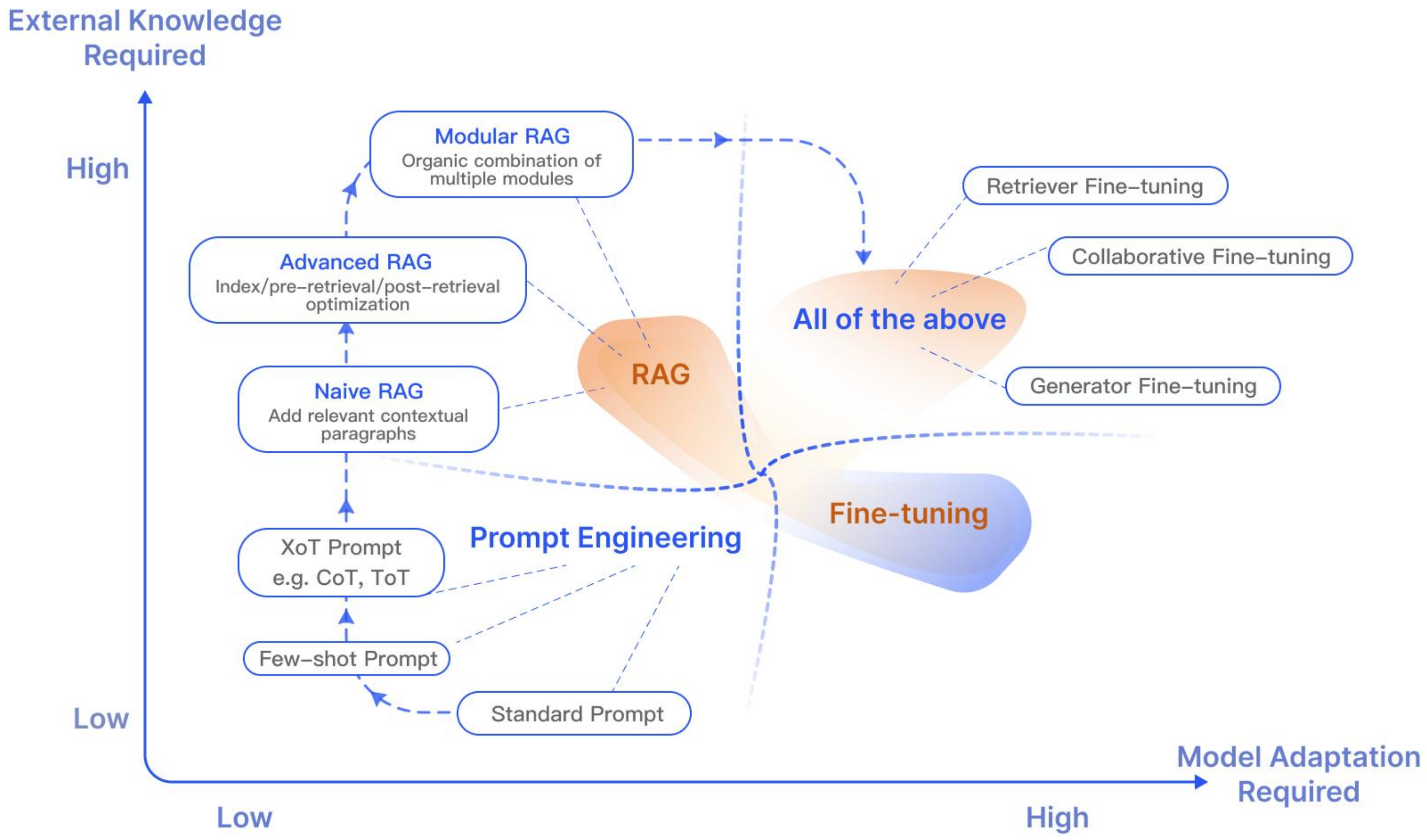
Symbolic Knowledge or Parametfic Knowledge

Ways to optimize LLMs.

Prompt Engineering

Retrieval-Augmented Generation

Instruct / Fine-tuning



A typical case of RAG

RAG vs Fine-tuning

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.

RAG Applications

Scenarios where RAG is applicable:

- Long-tail distribution of data
- Frequent knowledge updates
- Answers requiring verification and traceability
- Specialized domain knowledge
- Data privacy preservation

Q&A
RETRO (Borgeaud et al2021)
REALM (Gu et al, 2020)
ATLAS (lzacard et al, 2023)

Fact Checking
RAG (Lewis et al, 2020)
ATLAS (lzacard et al, 2022)
Evi. Generator (Asai et al, 2022a)

Dialog
BlenderBot3 (Shuster et al.2022)
Internet-augmented generation
(Komeili et a., 2022)

Summary
FLARE (Jiang et al, 2023)

Machine Translation
kNN-MT (Khandelwal et al., 2020)TRIME-MT (Zhong et al., 2022)

Code Generation
DocPrompting (Zhou et al., 2023
Natural ProverWelleck et al., 2022)

Natural Language Inference
kNN-Prompt (Shi et al., 2022)
NPM (Min et al., 2023)

Sentiment analysis
kNN-Prompt (Shi et al., 2022)NPM (Min et al., 2023)

Commonsense reasoning
Raco (Yu et al, 2022)

PART 02

RAG Paradigms Shifting

Naive RAG

Step1 Indexing

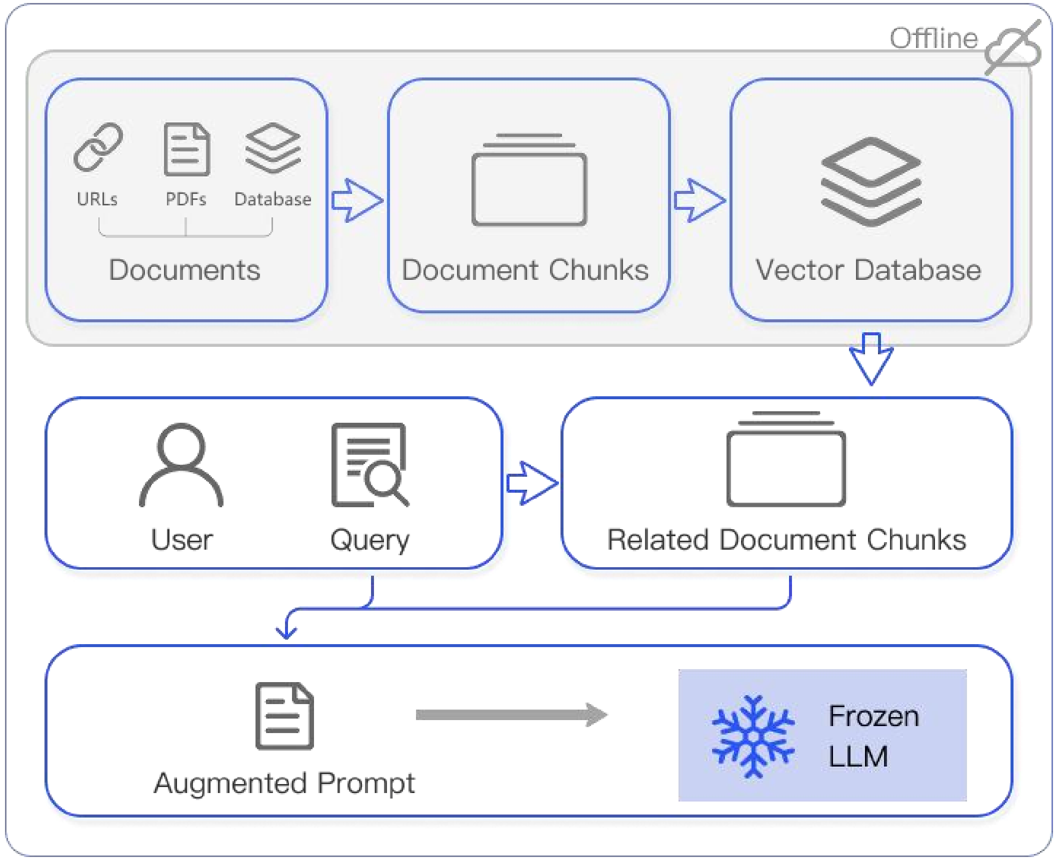
1. Divide the document into even chunks, each chunk being a piece of the original text.
2. Using the encoding model to generate an embedding for each chunck.
3. Store the Embedding of each block in the vector database.

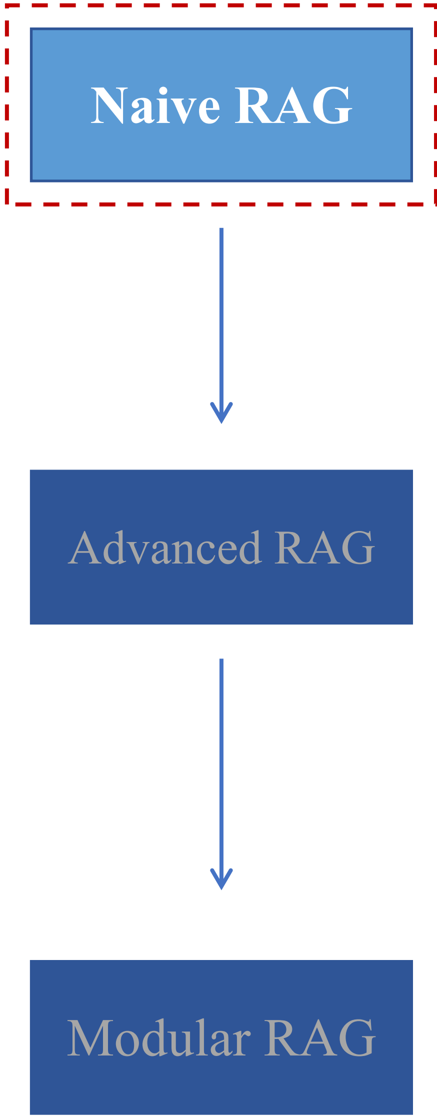
Step2 Retrival

Retrieve the k most relevant documents using vector similarity search.

Step3 Generation

The original query and the retrieved text are combined and input into a LLM to get the final answer





Advanced RAG

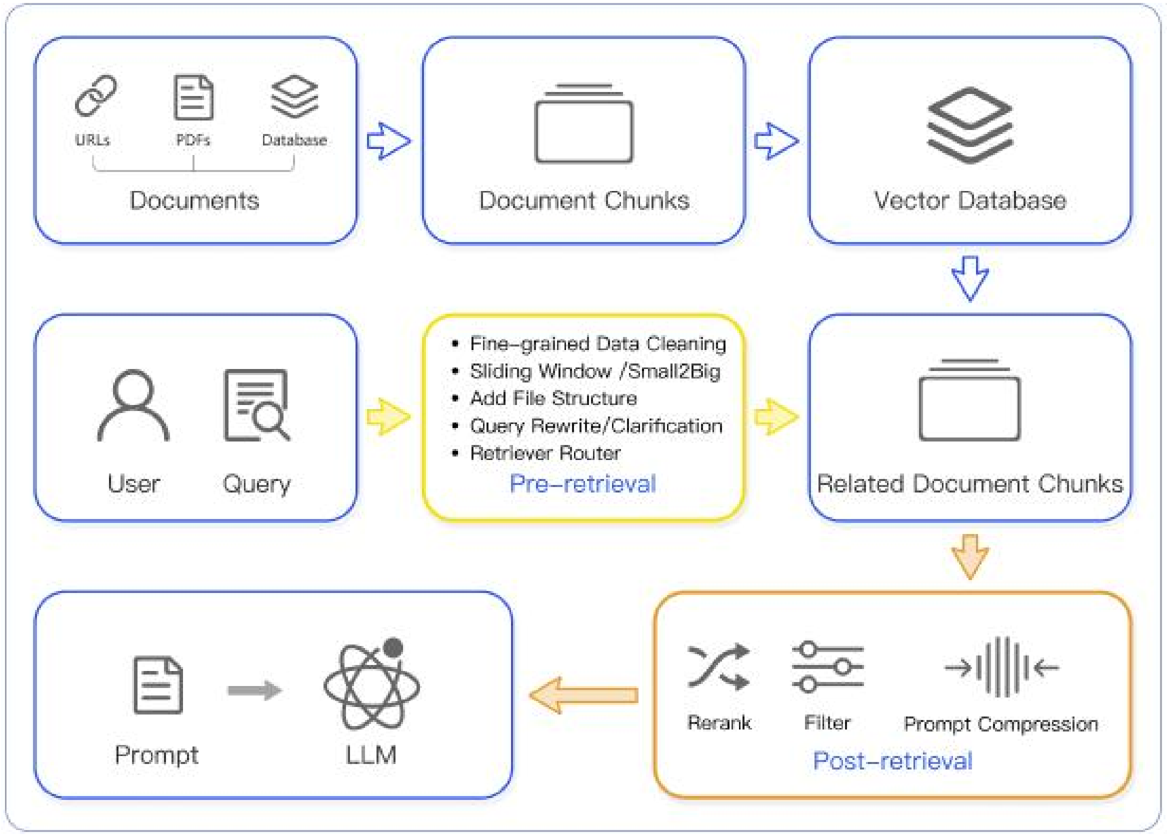
Index Optimization → Pre-Retrieval Process → Retrieval → Post-Retrieval Process → Genaration

- **Optimizing Data Indexing:**

  sliding window, fine-grained segmentation、adding metadata

- **Pre-Retrieval Process:** retrieve routes, summaries, rewriting, and confidence judgment

- **Post-Retrieval Process:** reorder, filter content retrieval



Naive RAG

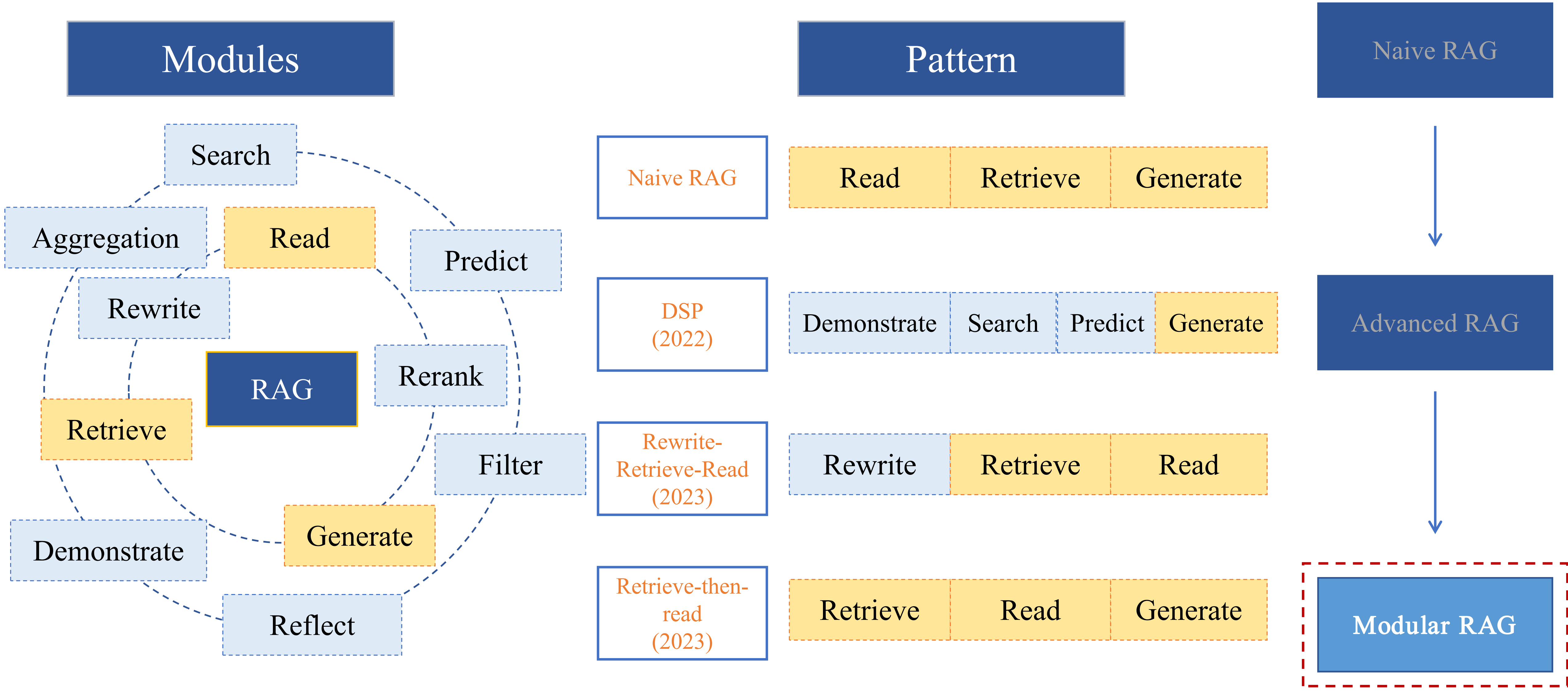
↓

Advanced RAG

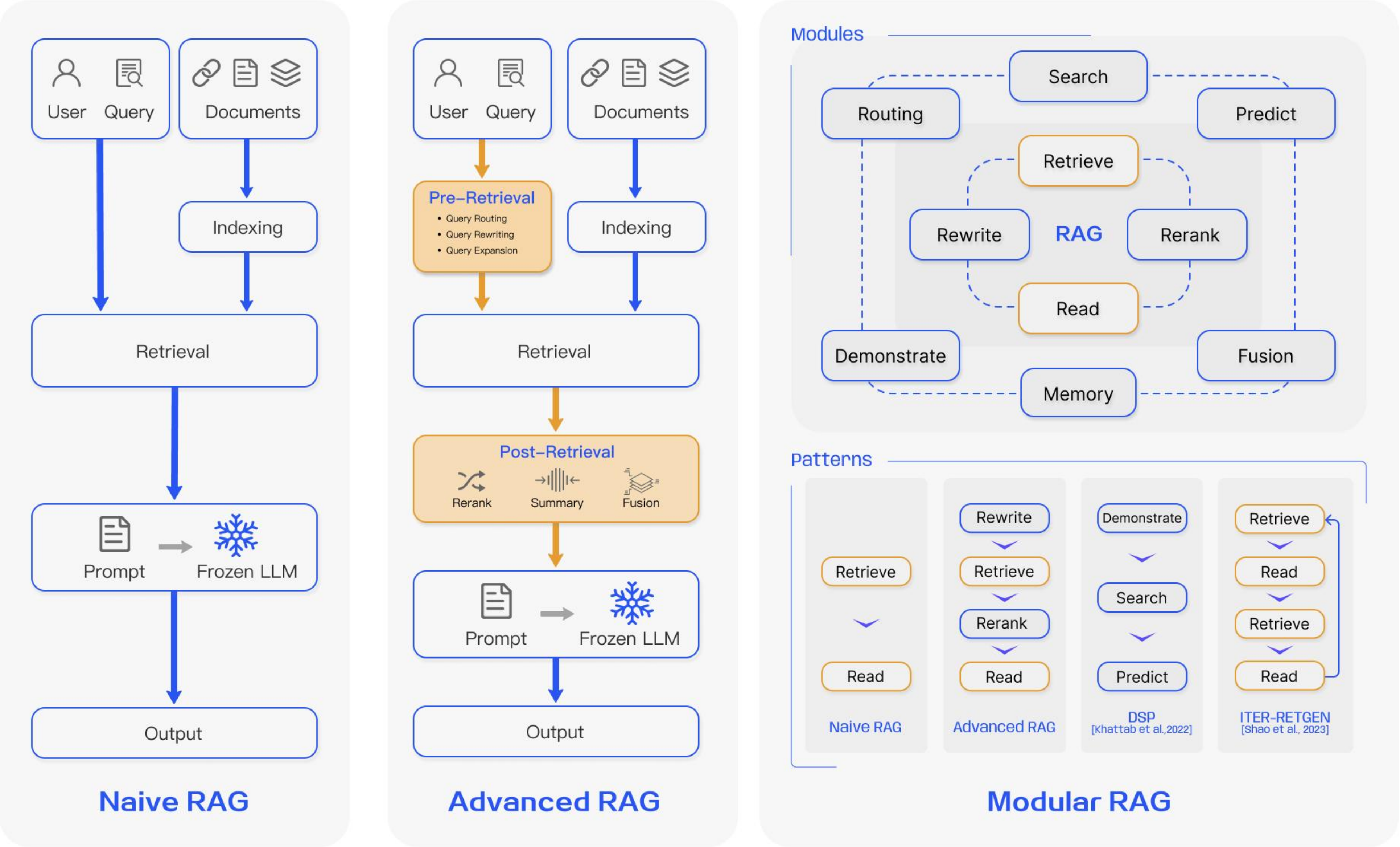
↓

Modular RAG

Modular RAG


Modules
Search
Aggregation
Read
Predict
Rewrite
RAG
Rerank
Retrieve
Filter
Generate
Demonstrate
Reflect
Pattern
Naive RAG
Read
Retrieve
Generate
DSP (2022)
Demonstrate
Search
Predict
Generate
Rewrite-Retrieve-Read (2023)
Rewrite
Retrieve
Read
Retrieve-then-read (2023)
Retrieve
Read
Generate
Naive RAG
Advanced RAG
Modular RAG

Comparison of RAG Paradigms



The three key questions of RAG

What to retrieve ?

- Token
- Phrase
- Chunk
- Paragraph
- Entity
- Knowledge graph

When to retrive ?

- Single search
- Each token
- Every N tokens
- Adaptive search

How to use the retrieved information ?

- Input/Data Layer
- Model/Intermediate Layer
- Output/Prediction Layer

Other Issues

Augmentation stage:

- Pre-training
- Fine-tuning
- Inference

Retrieval choice:

- BERT
- Roberta
- BGE
- ......

Model Collaboration

↔

↔

Scale selectionz

Generation choice:

- GPT
- Llama
- T5
- ......

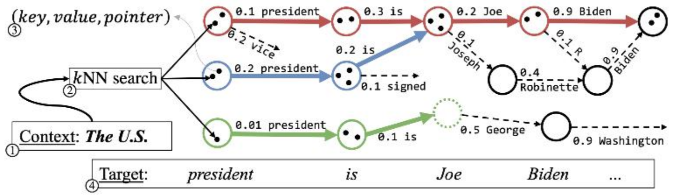
Key issue of RAG — What to retrieve

coarse

Retrieval granularity

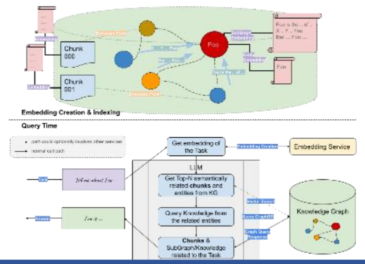
meticulous

Chunk | In-Context RAG 2023



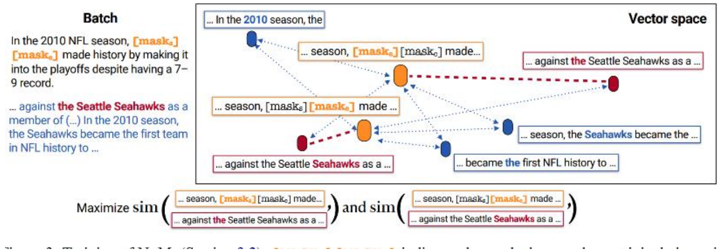
The search is broad, recalling a large amount of information, but with low accuracy, high coverage but includes much redundant information.

Knowledge Graph | 2023

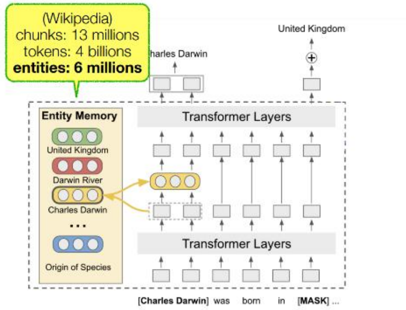


Richer semantic and structured information, but the retrieval efficiency is lower and is limited by the quality of KG.

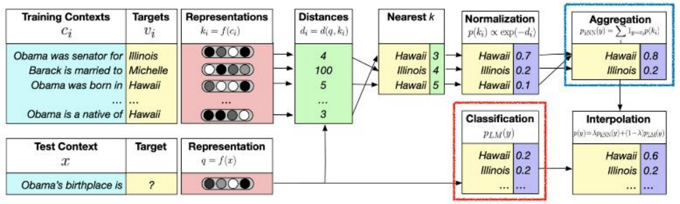
Phrase | NPM 2023



Entity | EasE 2022



Token | KNN-LMM 2019



It excels in handling long-tail and cross-domain issues with high computational efficiency, but it requires significant storage.

low

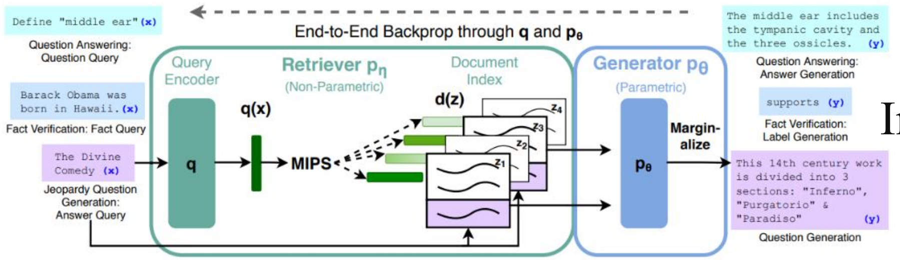
level of structuration

High

Key issue of RAG — How to use the retrieved content

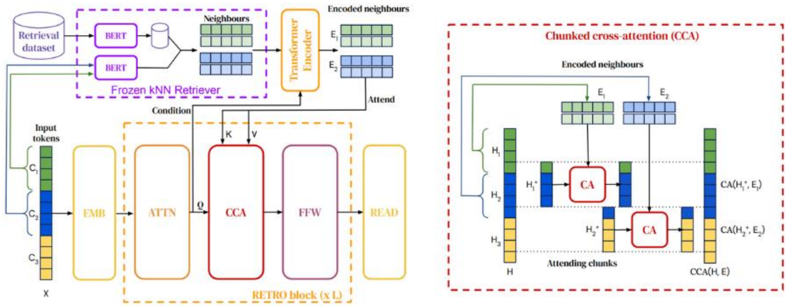
Integrating the retrieved information into different layers of the generation model,during inference process.

Integrate retrieval positions.



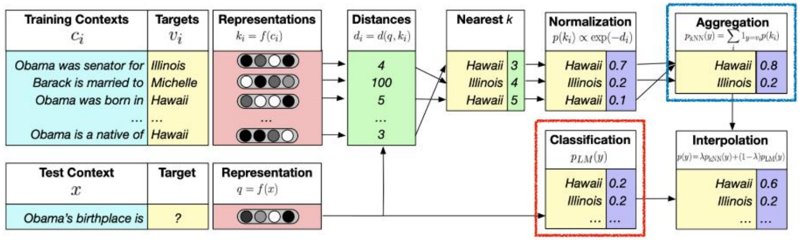
Input / Date layer

Using simple, but unable to support the retrieval of more knowledge blocks, and the optimization space is limited.



Model / Interlayer

Supports the retrieval of more knowledge blocks, but introduces additional complexity and must be trained.



Output /Prediction layer

Ensuring the output results are highly relevant to the retrieval content, but the efficiency is low.

Key issue of RAG — When to retrieve

High efficiency, but low relevance of the retrieved documents

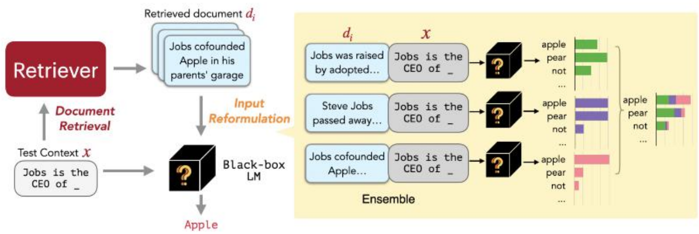
Balancing efficiency and information might not yield the optimal solution

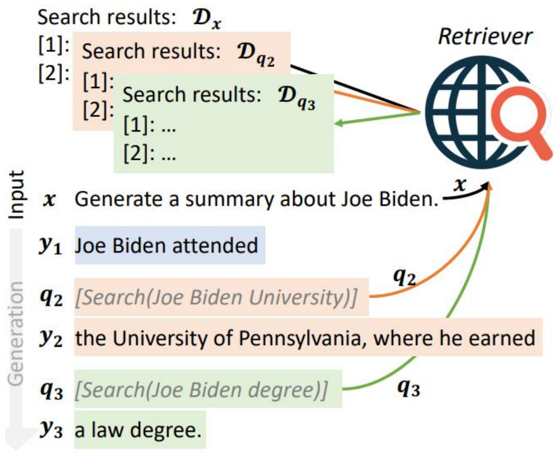
A large amount of information with low efficiency and redundant information.

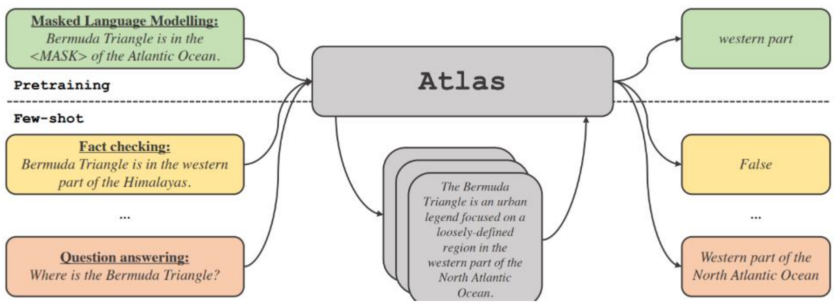
Once | Replug 2023

Adaptive | Flare 2023

Every N Tokens | Atlas 2023







Conducting once search during the reasoning process.

Adaptively conduct the search.

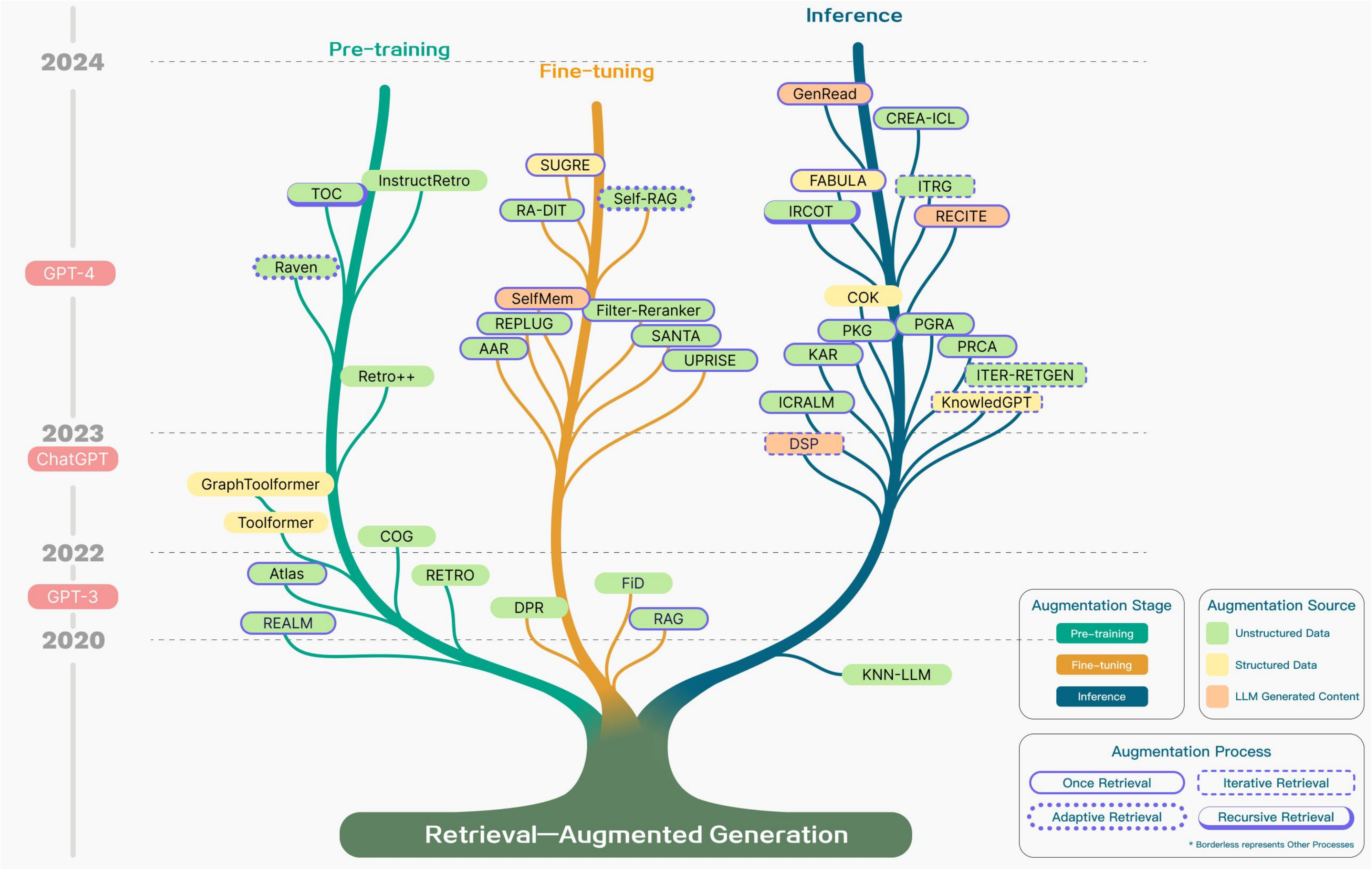
Retrieve once for every N tokens generated.

Low

Retrieval frequency

High

Overview of RAG Development



PART 03

Key Technologies and Evaluation

Techniques for Better RAG —— Data indexing optimization

Chunk Optimization

Small-2-Big

Embeding at sentence level expand the window during generation process.

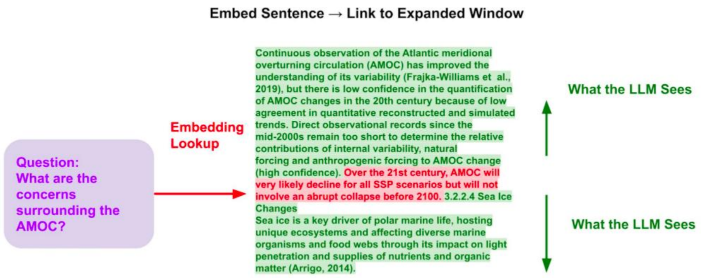
Slidingwindow

liding chunk covers the entire text, avoiding semantic ambiguity

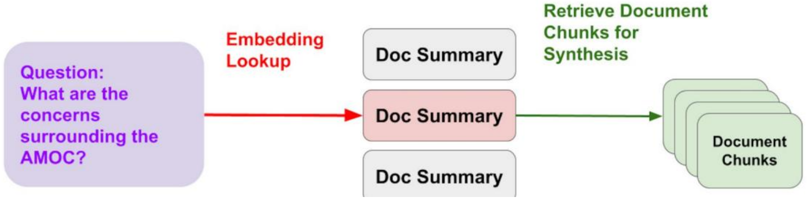
Summary

Retrieve documents through summaries, then retrieve text blocks from the documents.

Small-2-Big



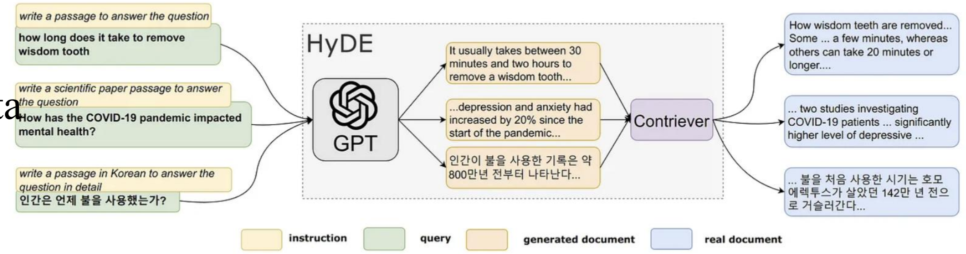
Abstract



Adding Metadata

Example | Page | Time | Type | Document Title

Pseudo Metadata



Metadata Filtering/Enrichment

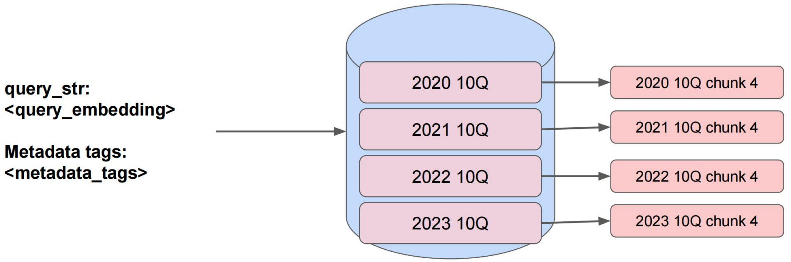
Pseudo Metadata Generation

Enhance retrieval by gener-ating a hypothetical document for the incoming query and creating qu-estions that the text block can answer.

Metadata filter

Dissect and annotate the document. During the query, infer metadata filters in addition to semantic queries

Metadata filter

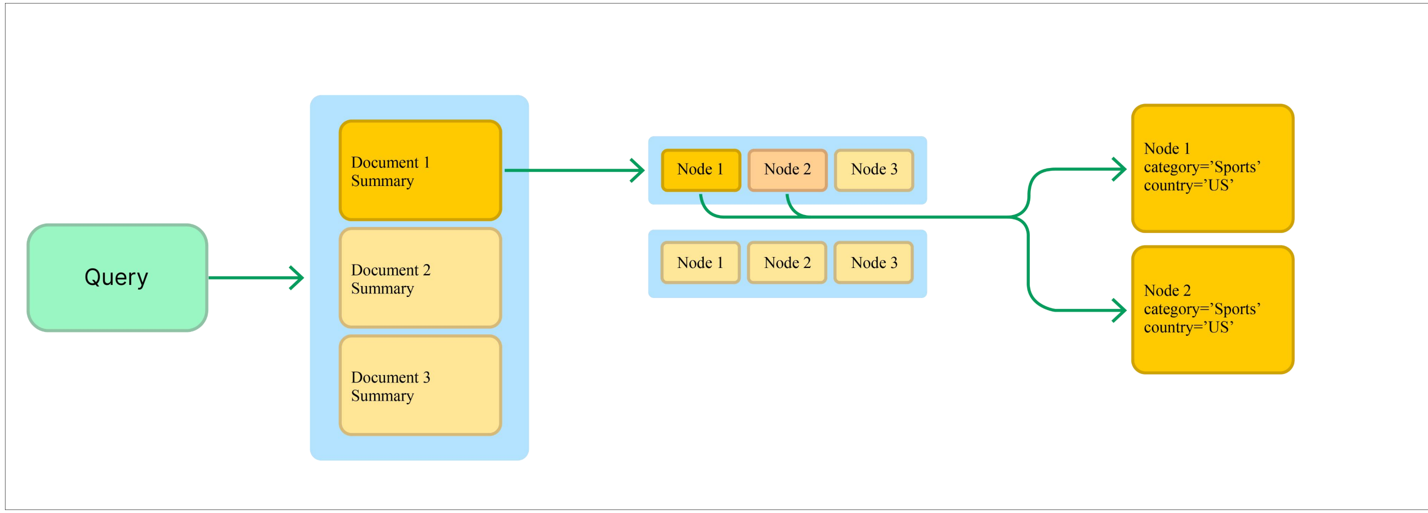


Techniques for Better RAG —— Structured Corpus

Hierarchical Organization of Retrieval Corpora

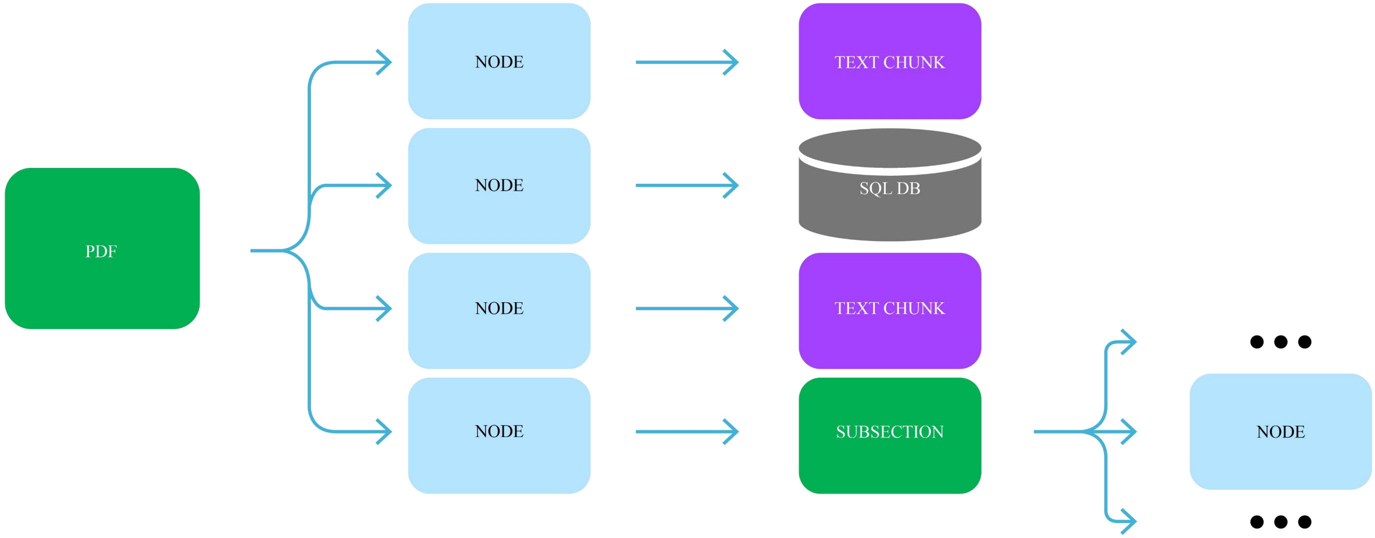
- Summary → Document

Replace document retrieval with summary retrieval, not only retrieving the most directly relevant nodes, but also exploring additional nodes associated with those nodes.



- Document → Embedded Objects

Documents have embedded objects (such as tables, charts), first retrieve entity reference objects, then query underlying objects, such as document blocks, databases, sub-nodes.

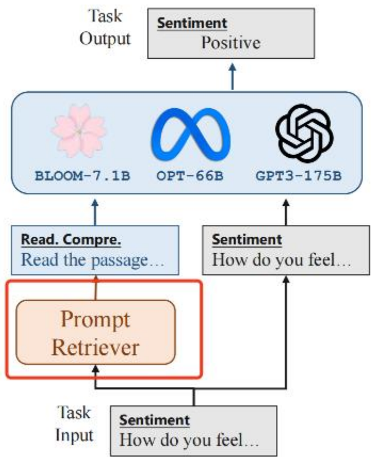


Techniques for Better RAG —— Retrieval Source Optimization

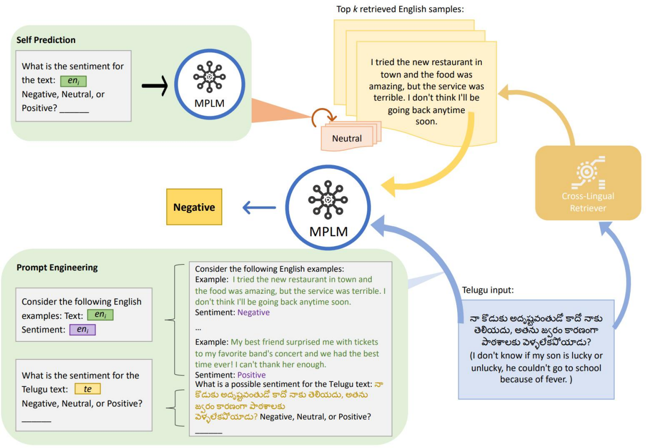
Unstructured Data
- Phrases
- Prompt
- Cross-lingustic

Structured Data
- Triples
- Subgraphs

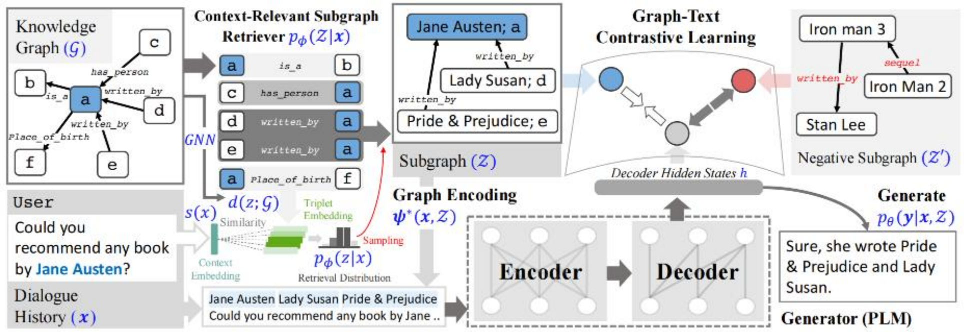
LLM
- LLM Memory
- Generated Text
- Generated Code



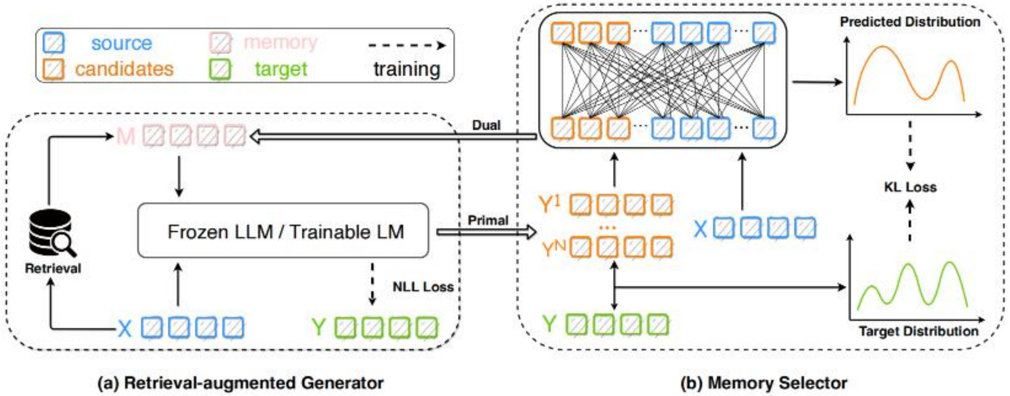
Prompt | UPRISE [Cheng et al.,2023]



Cross-language| CREA-ICL [Li et al., 2023]



Subgraph | SUGRE [Kang et al., 2023]



Memory | Selfmem [Cheng et al., 2023]

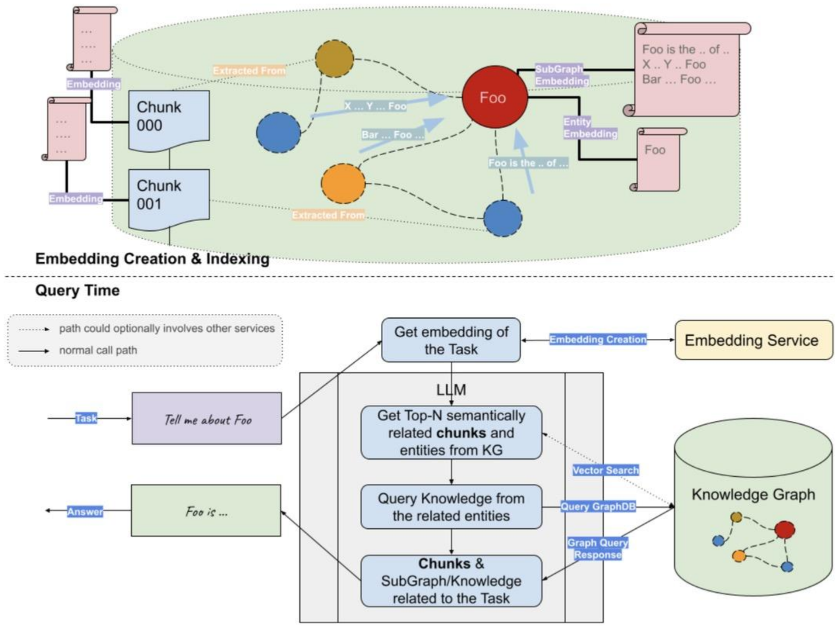
⏩ Techniques for Better RAG —— KG as a Retrieval Data Source

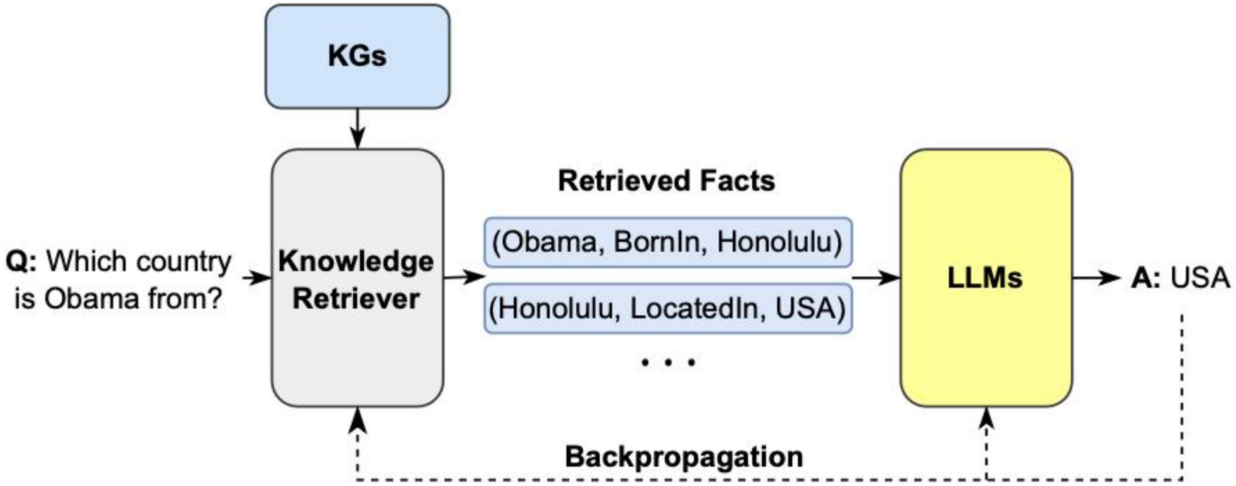
➢ GraphRAG

- ➢ Extract entities from the user's input query, then construct a subgraph to form context, and finally feed it into the large model for generation.

➢ Implementation

- ➢ Use LLM (or other models) to extract key entities from the question.
- ➢ Retrieve subgraphs based on entities, delving to a certain depth, such as 2 hops or even more.
- ➢ Utilize the obtained context to generate answers through LLM.

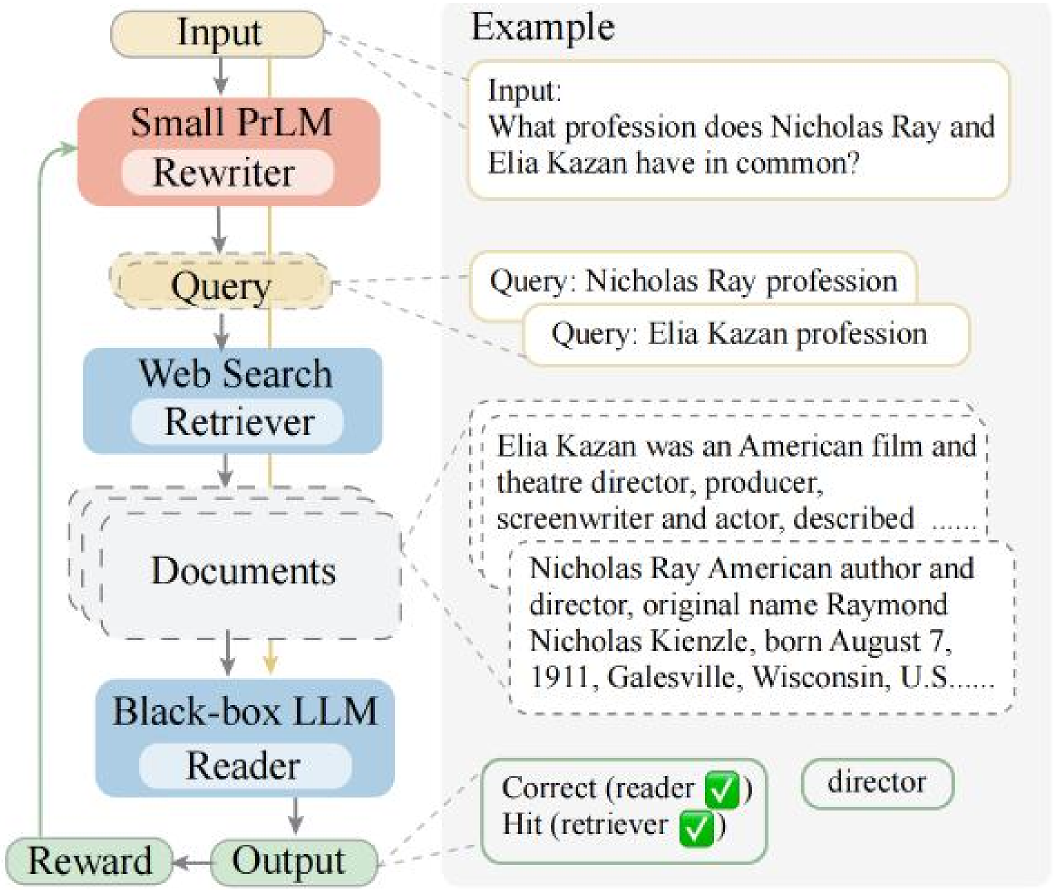




Techniques for Better RAG —— Query Optimization

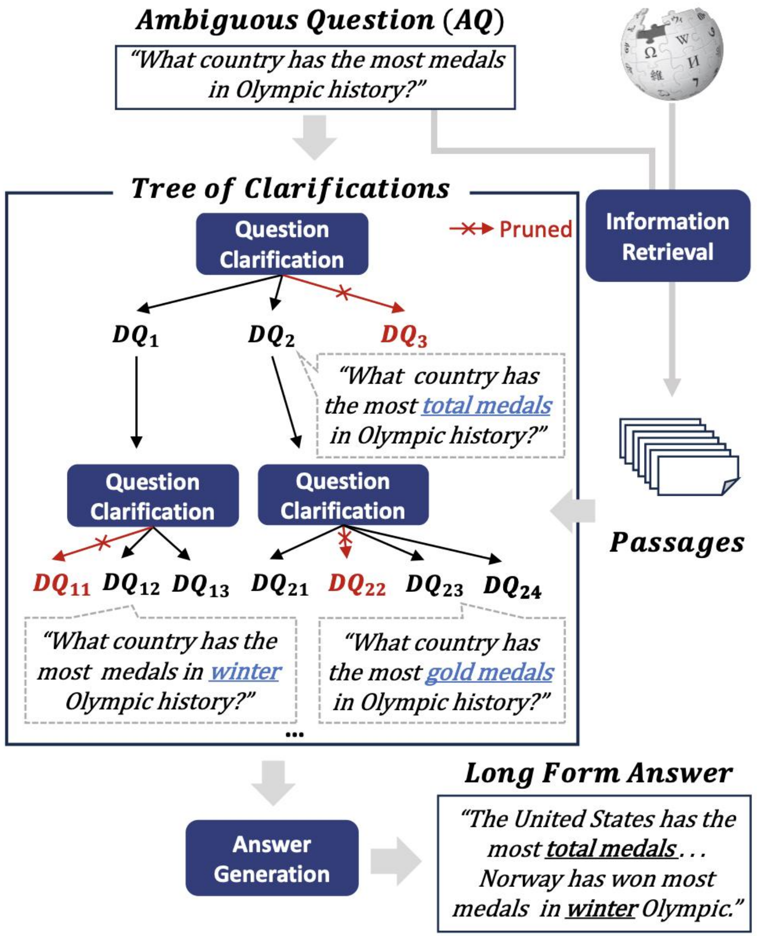
Questions and answers do not always possess high semantic similarity; adjusting the Query can yield better retrieval results.

Query Rewriting



Rewrite-Retrieve-Read [Ma et al., 2023]

Query Clarification

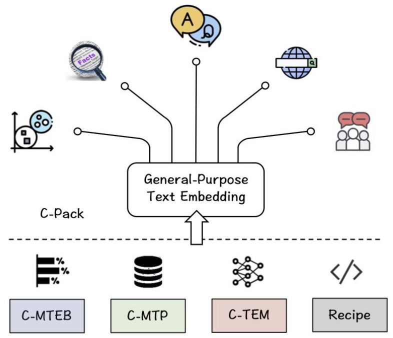


Tree of Clarifications（TOC）[Kim et al.,2023]

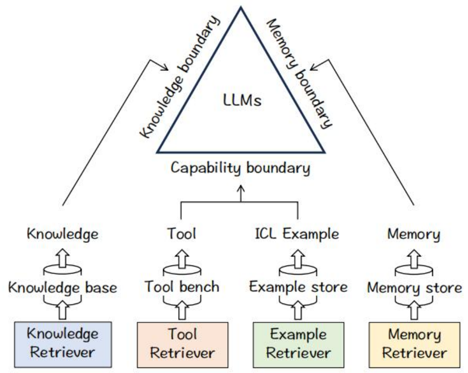
Techniques for Better RAG —— Embedding Optimization

Selecting a More Suitable Embedding Provider

cohere
M3E
VOYAGE AI
OpenAI
llmrails
BAAI 智源研究院

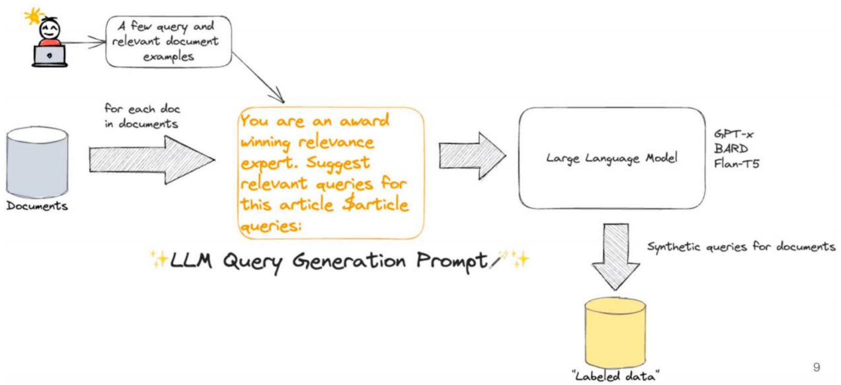


BAAI-General-Embedding (BGE)

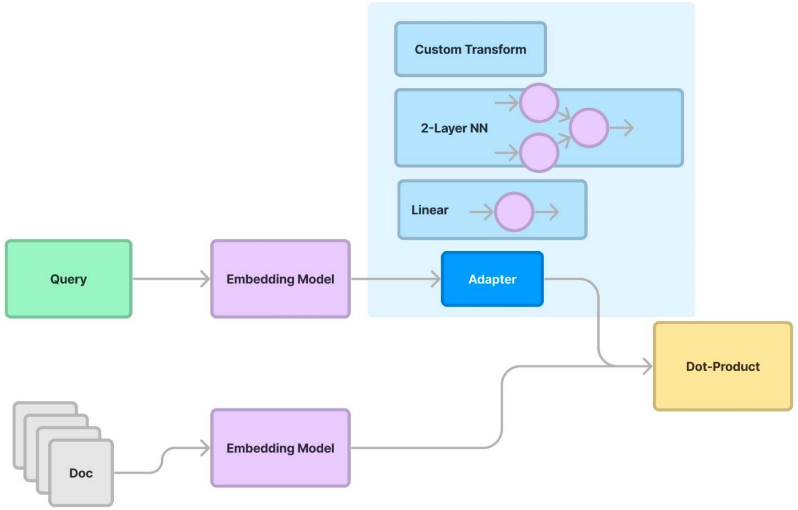


LLM-Embedder(BGE2) [Aksitov et al.,2023]

Fine-tuning the Embedding Model



Fine-tuning According to Domain-Specific Repositories and Downstream Tasks



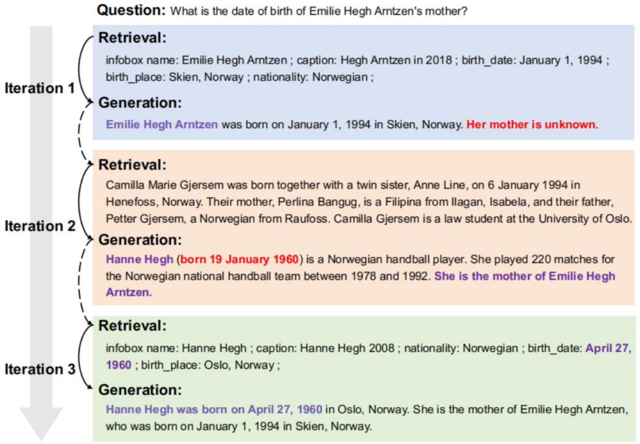
Fine-tuning the Adapter Module to Align the Embedding Model with the Retrieval Repository

Techniques for Better RAG —— Retrieval Process Optimization

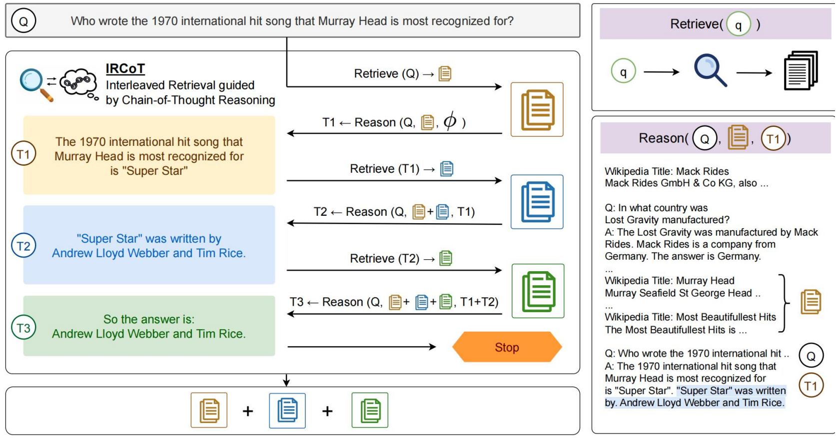
Iterative

Iteratively Retrieving from the Corpus to Acquire More Detailed and In-depth Knowledge

ITER [Feng et al., 2023]



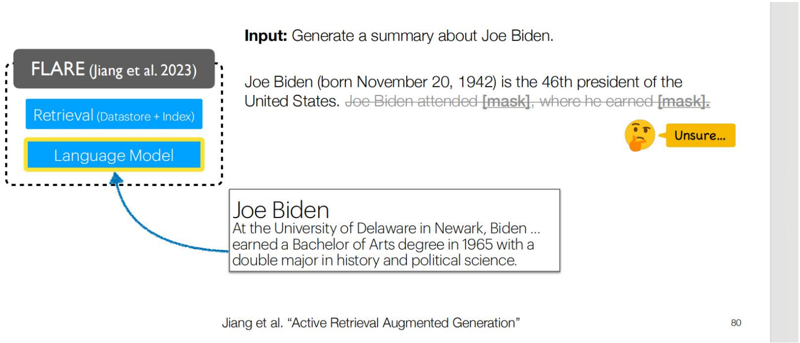
IRCOT [Trivedi et al.,2022]



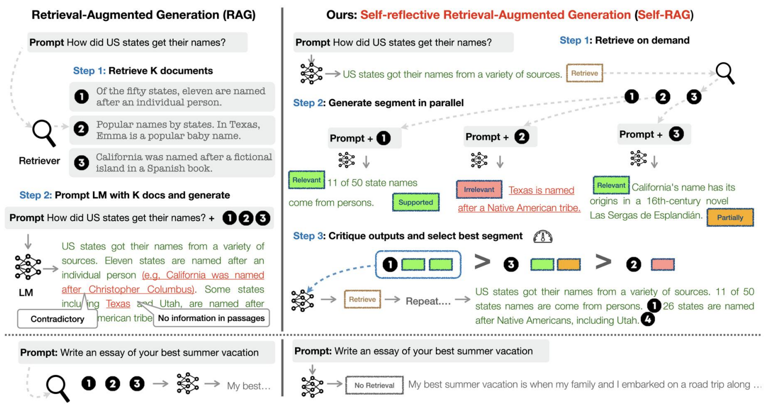
Adaptive

Dynamically Determined by the LLM, the Timing and Scope of Retrieva

FLARE [Jiang et al., 2023]

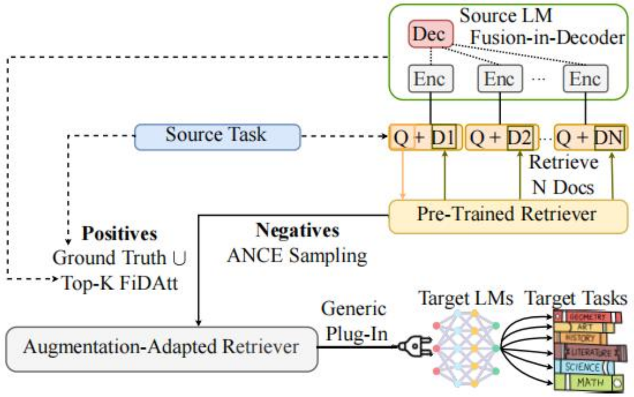


Self-RAG [Asai et al., 2023]



Techniques for Better RAG —— Hybrid (RAG + Fine-tuning)

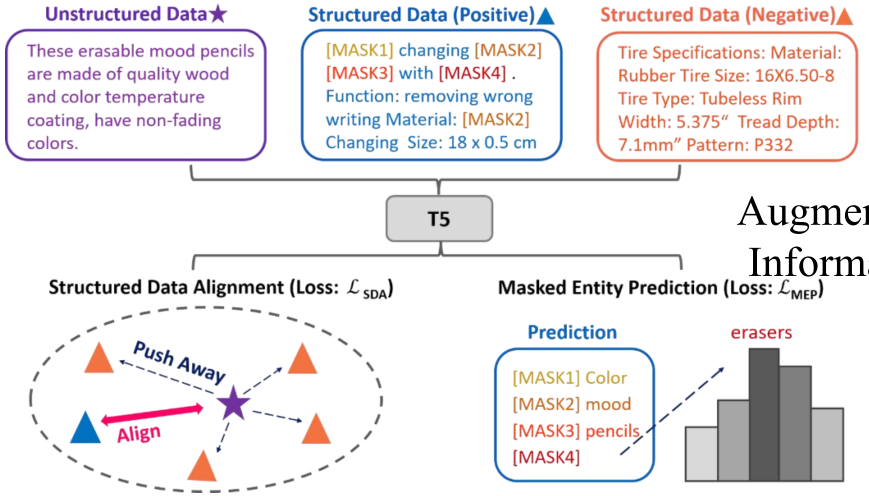
Retriever Fine-Tuning



Highly Adaptive General-Purpose Retrieval Plugin

AAR [Yu et al., 2023]

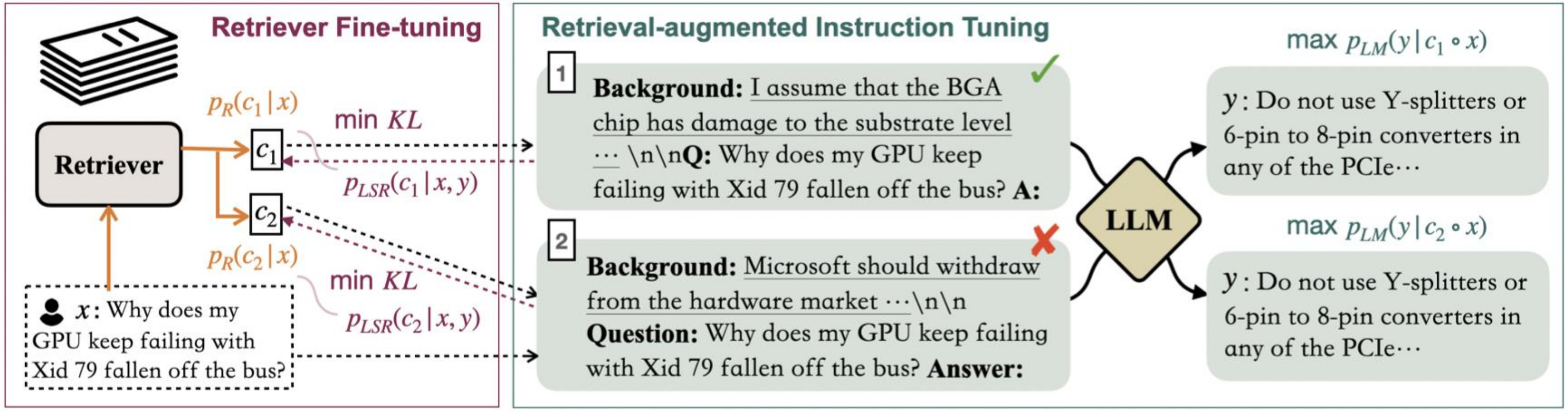
Generator Fine-Tuning



Augment with Structural Information Integration

SANTA [Li et al., 2023]

Collaborative Fine-Tuning



RA-DIT [Lin et al., 2023]

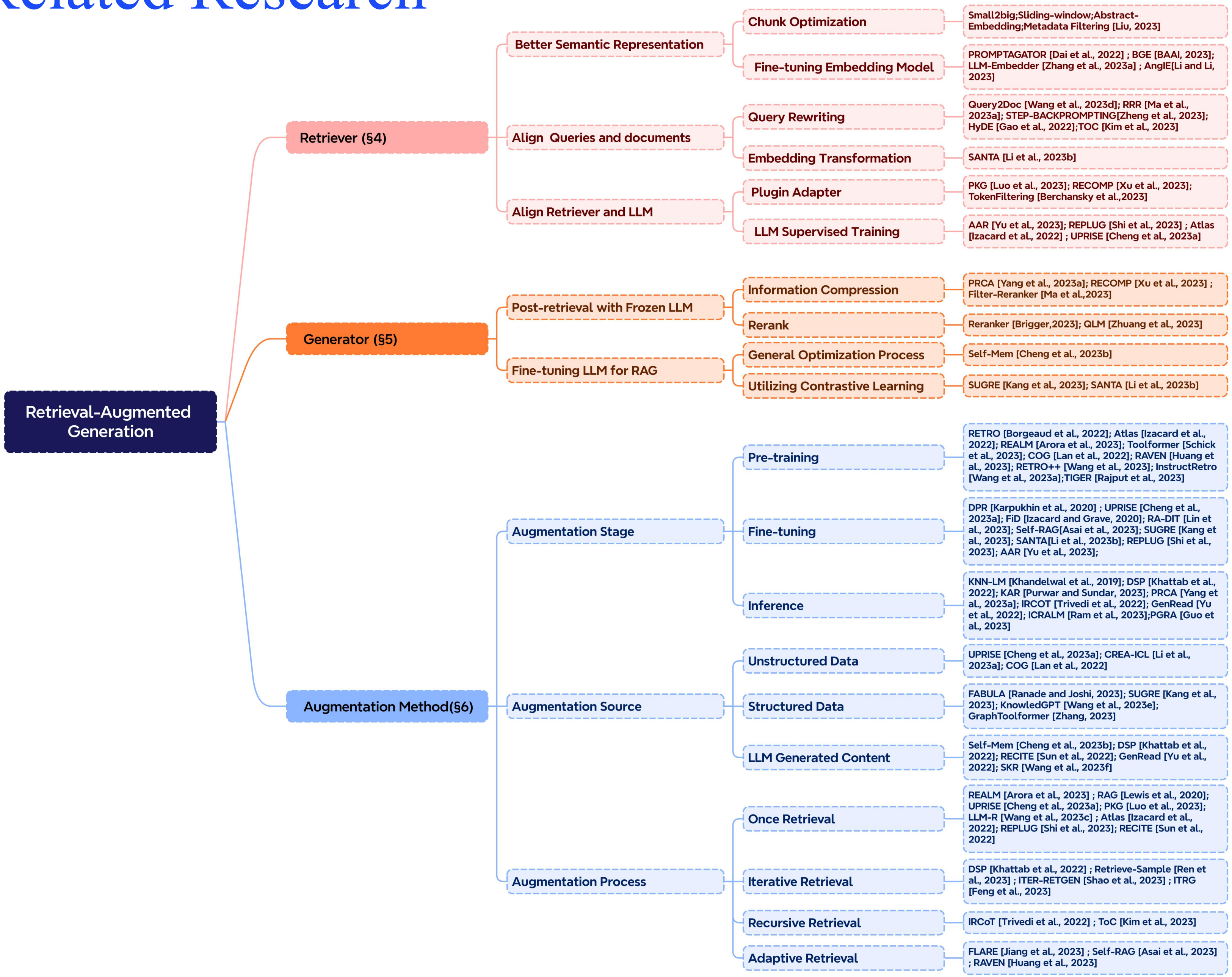
- R-FT

Minimizing the KL Divergence Between the Retriever Distribution and LLM Preferences

- LM-FT

Maximizing the Likelihood of the Correct Answer Given Retrieval-Augmented Instructions

Summary of Related Research



《Retrieval-Augmented Generation for Large Language Models: A Survey》

How to Evaluate the Effectiveness of RAG

EvaluationMethods

Independent Evaluation

Retriever
Evaluate the Quality of Text Blocks Retrieved by the Query
Metrics: MRP, Hit Rate, NDCG

Generation/Synthesis
Quality of Context Enhanced with Retrieved Documents Evaluation
Metrics: Context Relevance

End-to-End Evaluation

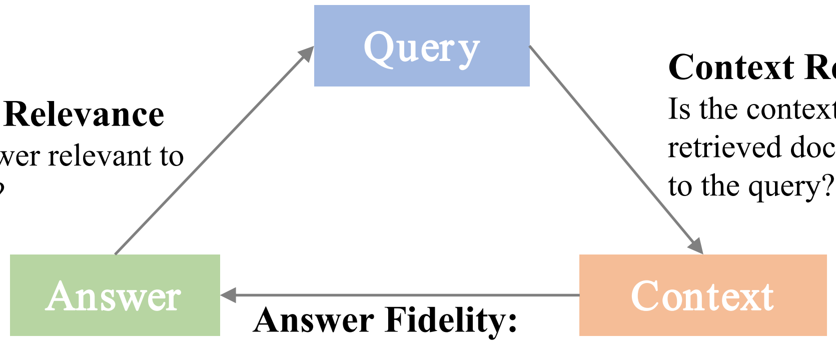
Evaluate the content ultimately generated by the model.

By generated conten
With labels： EM， Accuracy
Without labels: Fidelity, Relevance, Harmlessness

By evaluation method
Human evaluation
Automatic evaluation (LLM judge)

Key Metrics & Capabilities

Key Metrics



Answer Relevance
Is the answer relevant to the query?

Context Relevance:
Is the context enhanced with retrieved documents relevant to the query?

Answer Fidelity:
Is the answer based on the given context?

Key Capabilities

Noise Robustness
Can the model extract useful information from noisy documents?

Negative Rejection
When therequired knowledge is not exsiting in the retrieved documents, the answer should be refused.

Info Integration
Can the model answer complex questions that require integrating information from multiple documents?

Counterfactual Robustness
Can the model recognize the risk of known factual errors in the retrieved documents?

Assessment Framework

Use LLM as the adjudicator judge.

TruLens RAGAS — Based on handwritten prompt

ARES — Synthetic dataset + Fine-tuning + Ranking using confidence intervals

Evaluation →

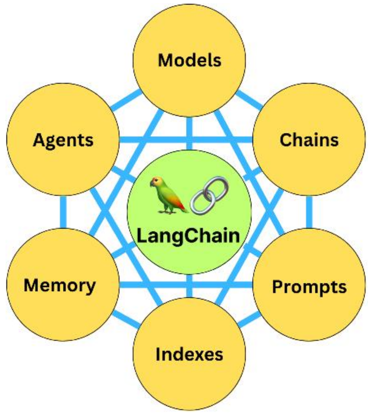
- Answer Fidelity
- Answer Relevance
- Contextual Relevance

PART 04

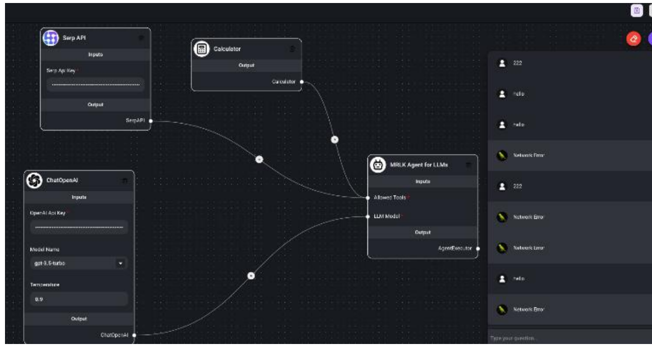
RAG Stack and Industry Practices

Existing Tech Stack for RAG

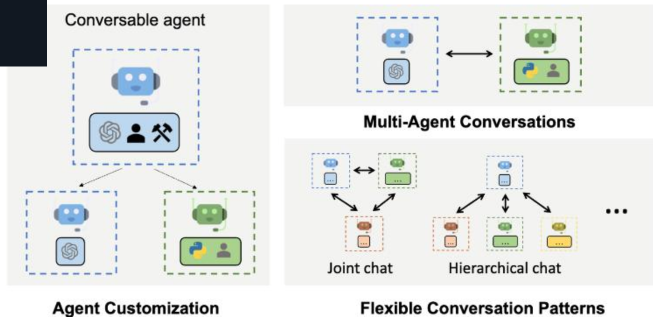
Name	Pros	Cons
LangChain	Modular, full-featured	Inconsistent behavior ,API conceals details,complexity and low flexibility.
LlamaIndex	Focus on RAG	Requires combination use, low customization.
FlowiseAI	Easy to get started, visualized workflows.	Does not support complex scenarios.
AutoGen	Adapts to multi-agent scenarios.	Low efficiency, requires multiple rounds of dialogue.



LangChain


FlowiseAI


LlamaIndex



AutoGen

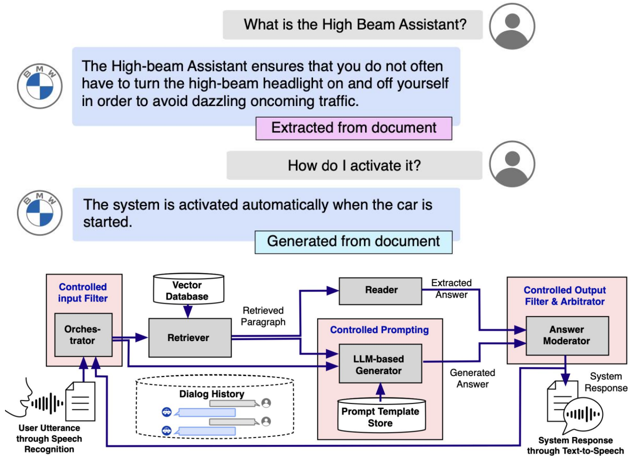
RAG Industry Application Practices

NetEase - ChatBI

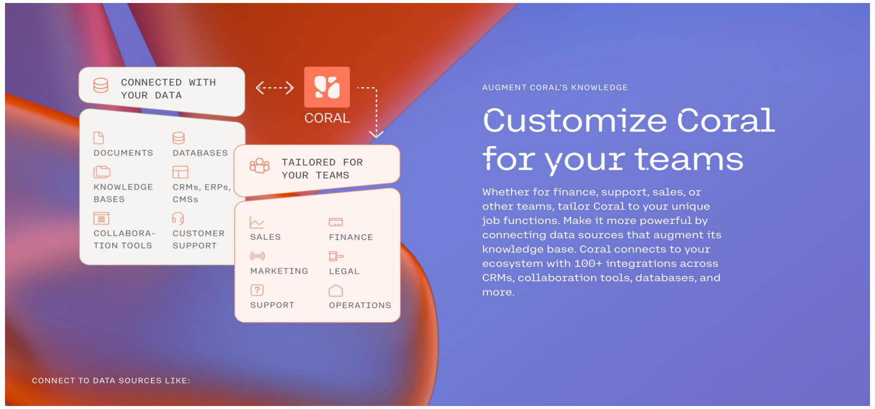
The intelligent upgrade of traditional industries

RAG

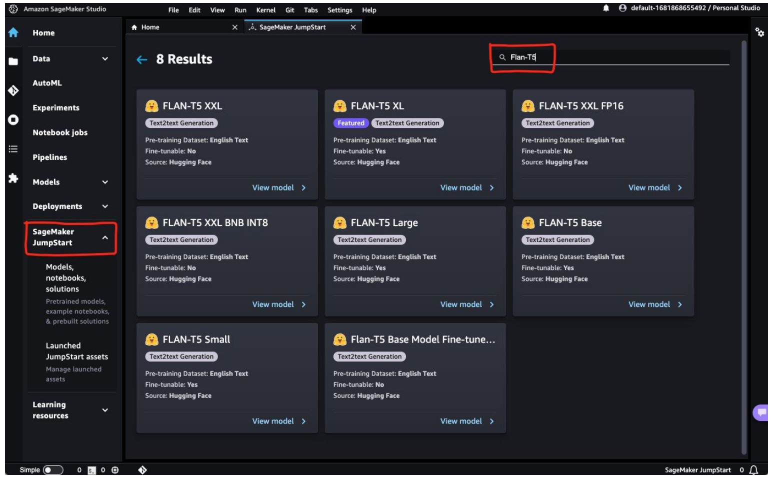
AI Toolchain Enhancement



BMW - CarExpert



Cohere - Coral

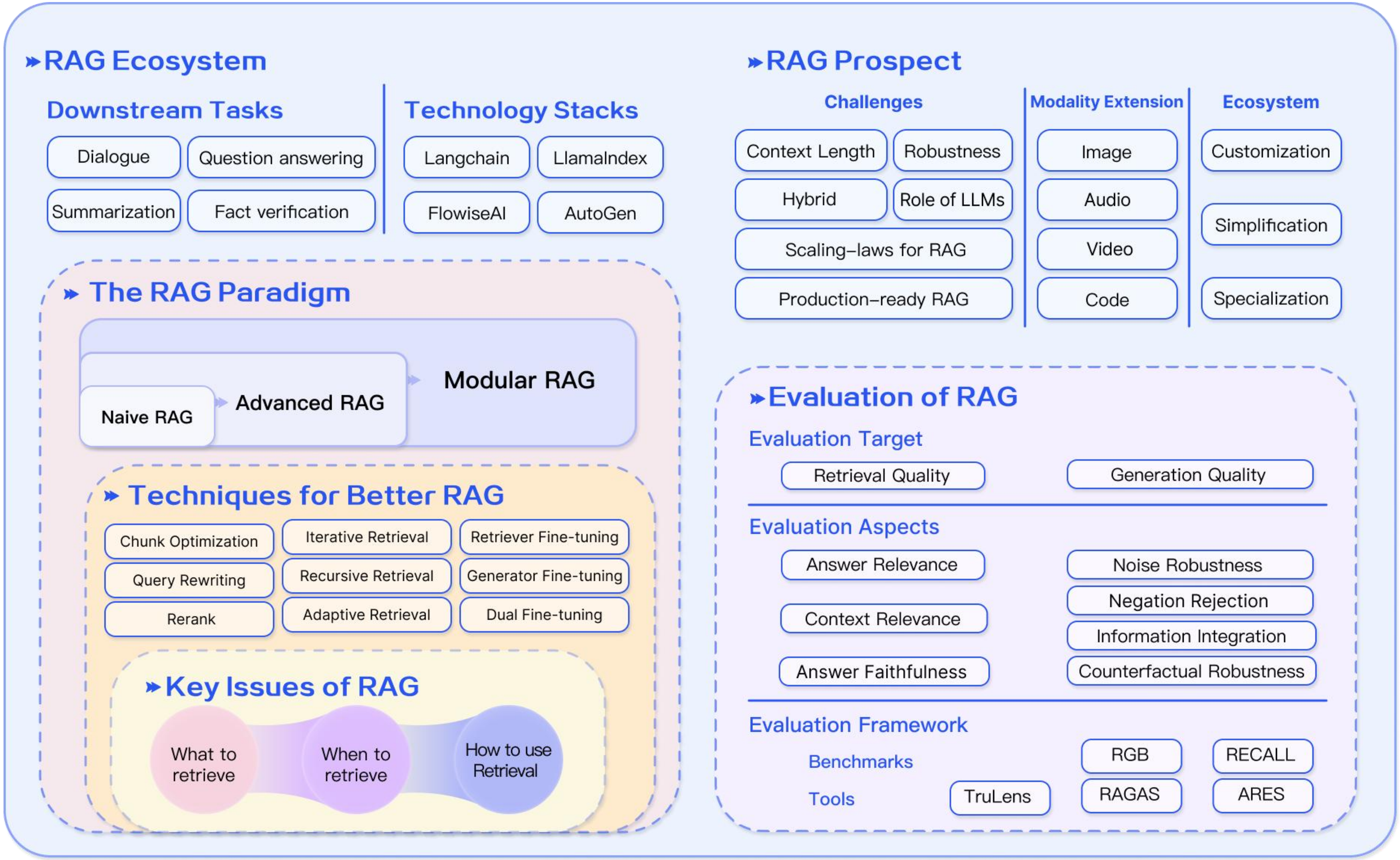


Amazon - Kendra

PART 06

Summary and Outlook

Summary — The Framework of RAG



Summary —— Three Trends of RAG

RAG

Technology

- The Scaling Law of RAG Models
- How to Improve the Efficiency of Retrieving Large-scale Data
- Mitigation of Forgetting in Long-context Scenarios
- Enhancement of Multimodal Retrieval

Paradigm

- Modularity Will Become Mainstream
- Patterns for Module Organization Await Refinement
- Evaluation Systems Need to Evolve and Improve with Time

Ecological Environment

- Preliminary Formation of Toolchain Technology Stack
- One-stop Platform Still Requires Polishing
- Explosion of Enterprise-level Applications

Prospects — Existing Challengs of RAG

Further address the challenges faced by RAG itself

Long context

- Retrieved content is excessive, exceeding window limit.
- The context is too long to result Lost in the Milddle.
- If the context window is not limited, is there still a need for RAG?

Coordination with FT

- How to simultaneously leverage the effects of RAG and FT.
- How do the two coordinate, how are they organized, is it in Pipeline, alternating, or end-to-end?

The role of LLMs

- LLM can be used for retrieval (LLM generation replaces retrieval, retrieving from LLM memory), for generation, and for evaluation. How to further explore the potential of LLM in RAG.

Robustness

- How to handle the incorrect content retrieved
- How to filter and verify the content retrieved.
- How to improve the model's resistance to toxicity and noise

Scaling Law

- Does the RAG model satisfy the Scaling Law
- Does RAG exhibit, or under what scenarios does it exhibit an Inverse Scaling Law

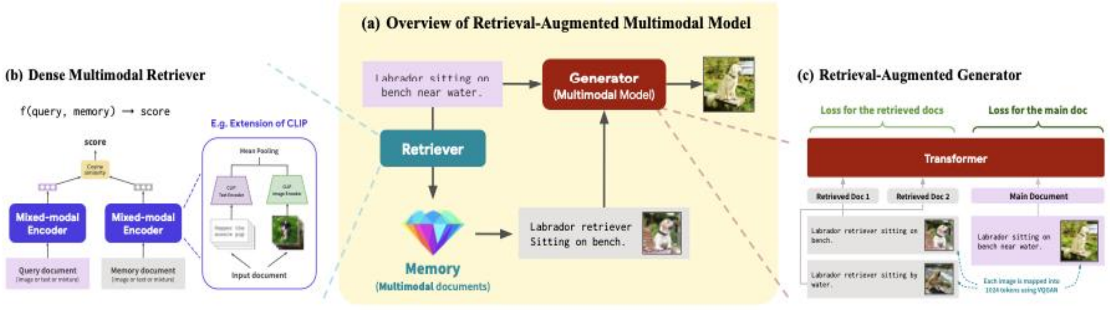
Engineering Practice

- How to reduce the latency of retrieving ultra-large-scale corpora.
- How to ensure that the content retrieved is not leaked by large models

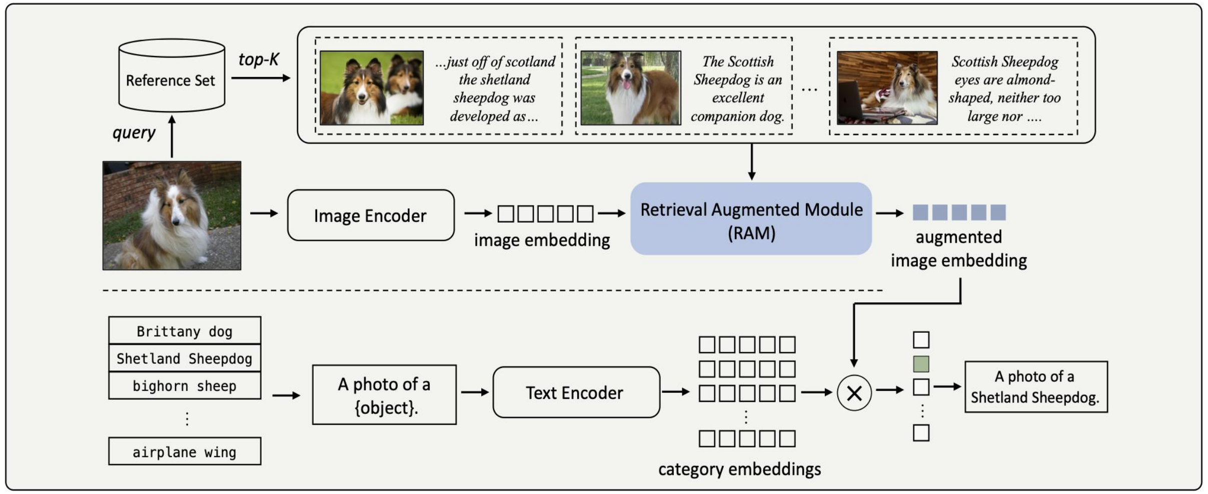
Prospects — Mult-Modality Extension

Transferring the concept of RAG from text to other modalities of data

Image

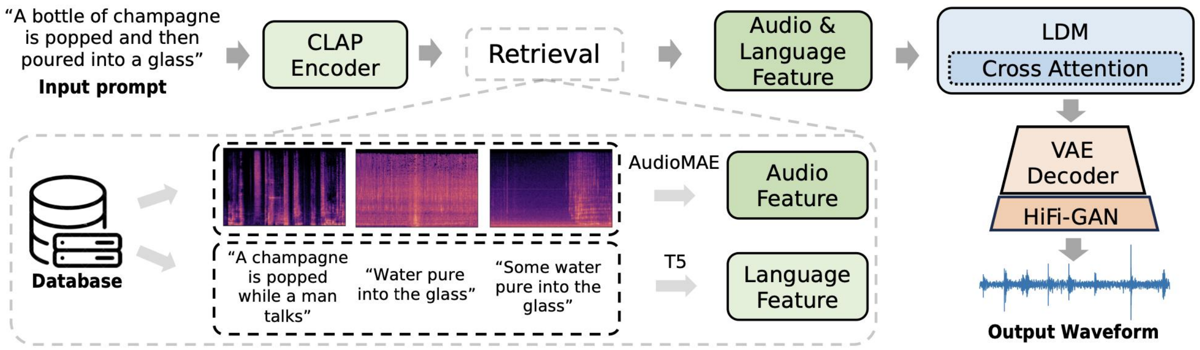


RA-CM3 [Yasunaga et al.,2023]



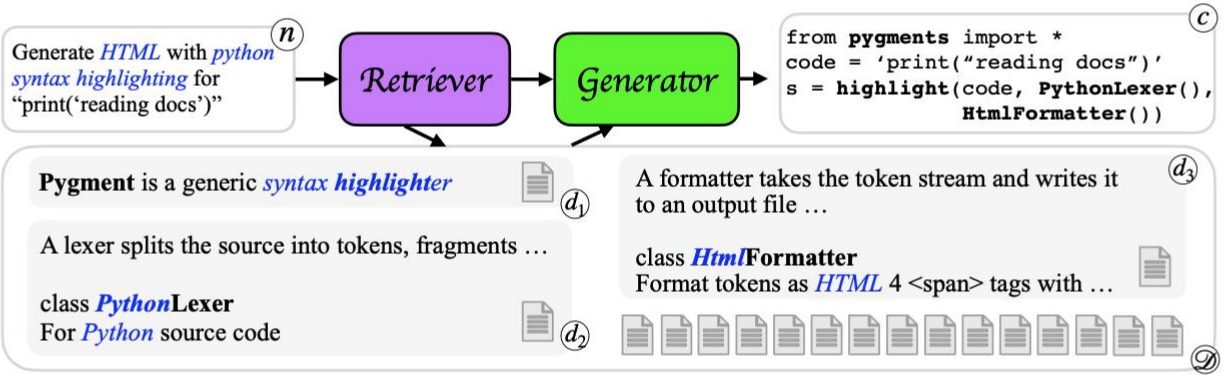
RA-CLIP [Xie et al.,2023]

Vedio



Re-AudioLDM [Yuan et al.,2023]

Code

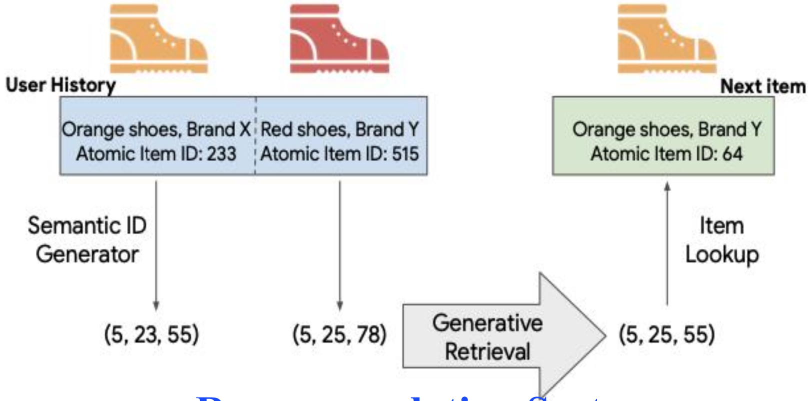


DocPrompting [Zhou et al.,2023]

Prospects —— Development of RAG Ecosystem

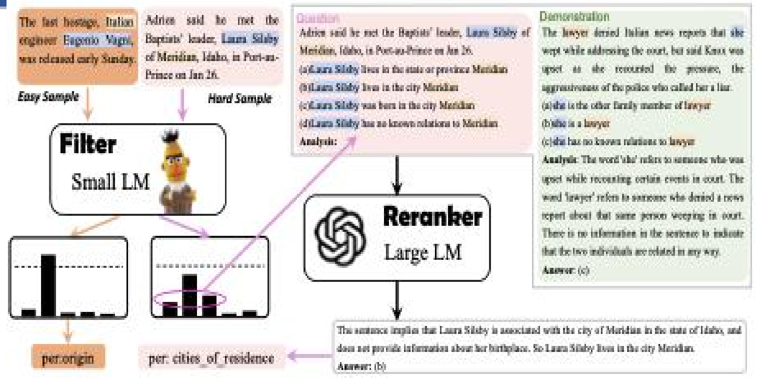
Further expand the downstream tasks of RAG and improve ecological construction

Downstream Task Development and Evaluation



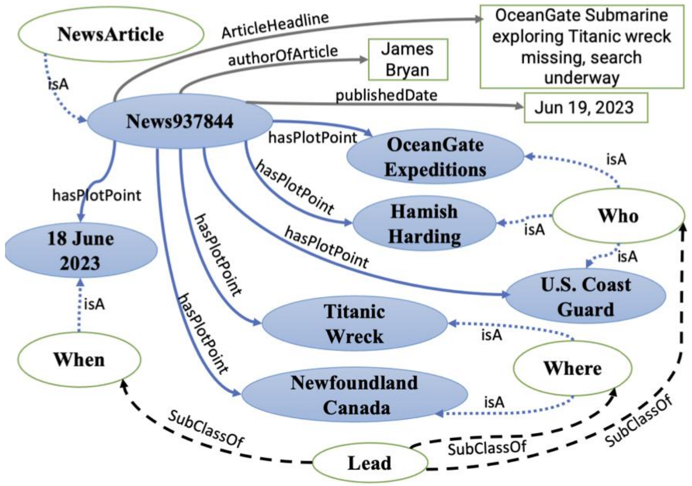
Recommendation System

| TIGER [Rajput et al.,2023]



Information extraction

| Filter- Rerank [Ma et al.,2023]



Report generation

| FABULA [Ranade et al.,2023]

Technology Stack Construction

- Customized function, meeting a variety of needs
- Simplified use, further reducing the barrier to entry.
- Specialized functions, gradually towards production environments.

Verba
The Golden RAGtriever

Personal Knowledge Assistant Based on RAG

haystack
by deepset

Open-source framework for production environments

Reference

1. Alon, U. et al. Neuro-Symbolic Language Modeling with Automaton-augmented Retrieval.
2. Lewis, P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.
3. Guu, K., Lee, K., Tung, Z., Pasupat, P. & Chang, M.-W. REALM: Retrieval-Augmented Language Model Pre-Training. Preprint at http://arxiv.org/abs/2002.08909 (2020).
4. Dai, Z. et al. Promptagator: Few-shot Dense Retrieval From 8 Examples. Preprint at http://arxiv.org/abs/2209.11755 (2022).
5. Izacard, G. et al. Atlas: Few-shot Learning with Retrieval Augmented Language Models. Preprint at http://arxiv.org/abs/2208.03299 (2022).
6. Gao, L., Ma, X., Lin, J. & Callan, J. Precise Zero-Shot Dense Retrieval without Relevance Labels. Preprint at http://arxiv.org/abs/2212.10496 (2022).
7. Muennighoff, N., Tazi, N., Magne, L. & Reimers, N. MTEB: Massive Text Embedding Benchmark. in Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics 2014–2037 (Association for Computational Linguistics, 2023).
8. Ren, Y. et al. Retrieve-and-Sample: Document-level Event Argument Extraction via Hybrid Retrieval Augmentation. in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) 293–306 (Association for Computational Linguistics, 2023).
9. Zhang, J. et al. ReAugKD: Retrieval-Augmented Knowledge Distillation For Pre-trained Language Models. in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) 1128–1136 (Association for Computational Linguistics, 2023). 10. Khattab, O. et al. Demonstrate-Search-Predict: Composing retrieval and language models for knowledge-intensive NLP. Preprint at http://arxiv.org/abs/2212.14024 (2023).
11. Cheng, X. et al. Lift Yourself Up: Retrieval-augmented Text Generation with Self Memory. Preprint at http://arxiv.org/abs/2305.02437 (2023).
12. Luo, Z. et al. Augmented Large Language Models with Parametric Knowledge Guiding. Preprint at http://arxiv.org/abs/2305.04757 (2023).
13. Shi, W. et al. REPLUG: Retrieval-Augmented Black-Box Language Models. Preprint at http://arxiv.org/abs/2301.12652 (2023).
14. Yu, Z., Xiong, C., Yu, S. & Liu, Z. Augmentation-Adapted Retriever Improves Generalization of Language Models as Generic Plug-In. Preprint at http://arxiv.org/abs/2305.17331 (2023).
15. Kang, M., Kwak, J. M., Baek, J. & Hwang, S. J. Knowledge Graph-Augmented Language Models for Knowledge-Grounded Dialogue Generation. Preprint at http://arxiv.org/abs/2305.18846 (2023).
16. Trivedi, H., Balasubramanian, N., Khot, T. & Sabharwal, A. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. Preprint at http://arxiv.org/abs/2212.10509 (2023).
17. Wang, L., Yang, N. & Wei, F. Learning to Retrieve In-Context Examples for Large Language Models. Preprint at http://arxiv.org/abs/2307.07164 (2023).
18. Li, Z. et al. Towards General Text Embeddings with Multi-stage Contrastive Learning. Preprint at http://arxiv.org/abs/2308.03281 (2023).
19. Ng, Y. et al. SimplyRetrieve: A Private and Lightweight Retrieval-Centric Generative AI Tool. Preprint at http://arxiv.org/abs/2308.03983 (2023).
20. Huang, J. et al. RAVEN: In-Context Learning with Retrieval Augmented Encoder-Decoder Language Models. Preprint at http://arxiv.org/abs/2308.07922 (2023).

Reference

21. Zhu, Y. et al. Large Language Models for Information Retrieval: A Survey. Preprint at http://arxiv.org/abs/2308.07107 (2023).
22. Wang, X. et al. KnowledGPT: Enhancing Large Language Models with Retrieval and Storage Access on Knowledge Bases. Preprint at http://arxiv.org/abs/2308.11761 (2023).
23. Chen, J., Lin, H., Han, X. & Sun, L. Benchmarking Large Language Models in Retrieval-Augmented Generation. Preprint at http://arxiv.org/abs/2309.01431
24. Es, S., James, J., Espinosa-Anke, L. & Schockaert, S. RAGAS: Automated Evaluation of Retrieval Augmented Generation. Preprint at http://arxiv.org/abs/2309.15217 (2023).
25. Yoran, O., Wolfson, T., Ram, O. & Berant, J. Making Retrieval-Augmented Language Models Robust to Irrelevant Context. Preprint at http://arxiv.org/abs/2310.01558 (2023).
26. Feng, Z., Feng, X., Zhao, D., Yang, M. & Qin, B. Retrieval-Generation Synergy Augmented Large Language Models. Preprint at http://arxiv.org/abs/2310.05149 (2023).
27. Zheng, H. S. et al. Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models. Preprint at http://arxiv.org/abs/2310.06117 (2023).
28. Cheng, D. et al. UPRISE: Universal Prompt Retrieval for Improving Zero-Shot Evaluation. Preprint at http://arxiv.org/abs/2303.08518 (2023).
29. Wang, B. et al. InstructRetro: Instruction Tuning post Retrieval-Augmented Pretraining. Preprint at http://arxiv.org/abs/2310.07713 (2023).
30. Jiang, Z. et al. Active Retrieval Augmented Generation. Preprint at http://arxiv.org/abs/2305.06983 (2023).
31. Gou, Q. et al. Diversify Question Generation with Retrieval-Augmented Style Transfer. Preprint at http://arxiv.org/abs/2310.14503 (2023).
32. Ma, X., Gong, Y., He, P., Zhao, H. & Duan, N. Query Rewriting for Retrieval-Augmented Large Language Models. Preprint at http://arxiv.org/abs/2305.14283 (2023).
33. Yang, H. et al. PRCA: Fitting Black-Box Large Language Models for Retrieval Question Answering via Pluggable Reward-Driven Contextual Adapter. Preprint at http://arxiv.org/abs/2310.18347 (2023).
34. Kim, G., Kim, S., Jeon, B., Park, J. & Kang, J. Tree of Clarifications: Answering Ambiguous Questions with Retrieval-Augmented Large Language Models. Preprint at http://arxiv.org/abs/2310.14696 (2023).
35. Shao, Z. et al. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. Preprint at http://arxiv.org/abs/2305.15294 (2023).
36. Zhang, P., Xiao, S., Liu, Z., Dou, Z. & Nie, J.-Y. Retrieve Anything To Augment Large Language Models. Preprint at http://arxiv.org/abs/2310.07554 (2023).
37. Purwar, A. & Sundar, R. Keyword Augmented Retrieval: Novel framework for Information Retrieval integrated with speech interface. Preprint at http://arxiv.org/abs/2310.04205 (2023).
38. Lin, X. V. et al. RA-DIT: Retrieval-Augmented Dual Instruction Tuning. Preprint at http://arxiv.org/abs/2310.01352 (2023).
39. Yu, W. et al. Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models. Preprint at http://arxiv.org/abs/2311.09210 (2023).

Thank you!

For more information, please see:
Our paper : https://arxiv.org/abs/2312.10997
Our GitHub: https://github.com/Tongji-KGLLM/RAG-Survey