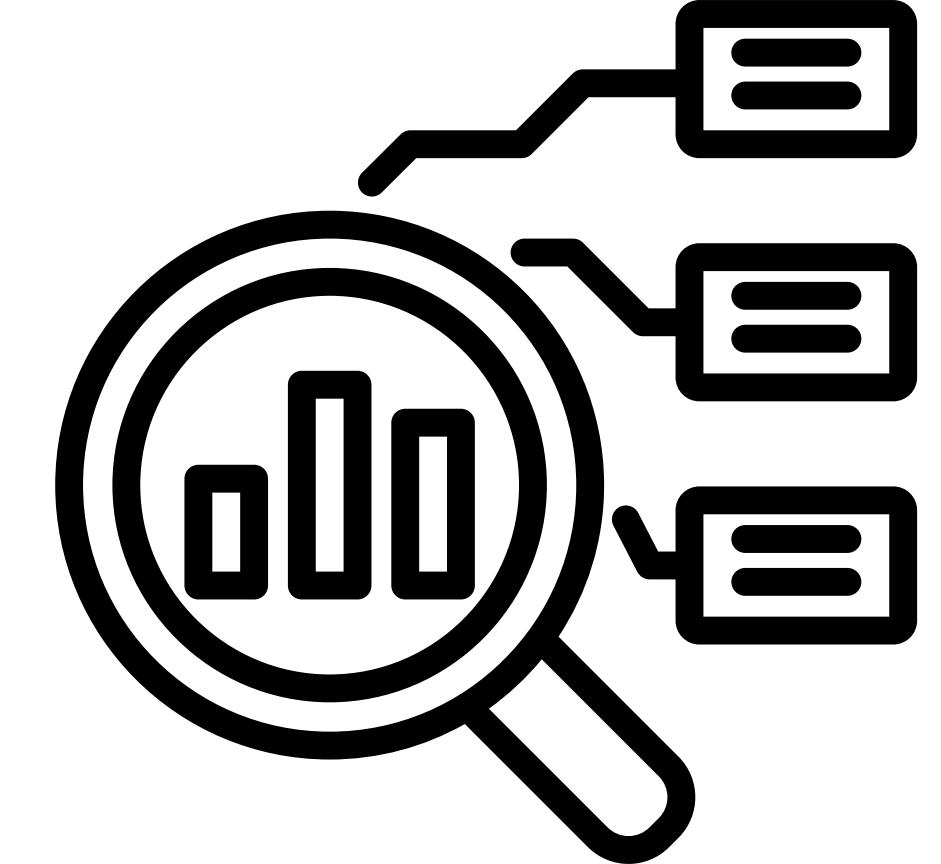


Evolution RAG Techniques 2025

By

Thanawich Juengkijthanawat (Guide)
polakorn anantapakorn (Ming)



Agenda

- **RAG-paradigm**
 - Navie RAG
 - Advanced RAG
 - Modular RAG
- **Core Retrieval Techiniques 2025**
 - **Search techniques**
 - Vector Search (Dense Retrieval)
 - Keyword Search (Sparse Retrieval)
 - Hybrid Search
- **Retrieval-Techniques**
 - **HyDE RAG**
 - **Apdative Retrieval**
 - **Long RAG**
 - **Graph RAG**
 - **(Summaries & Use case)**
- **Graph RAG DEMO**

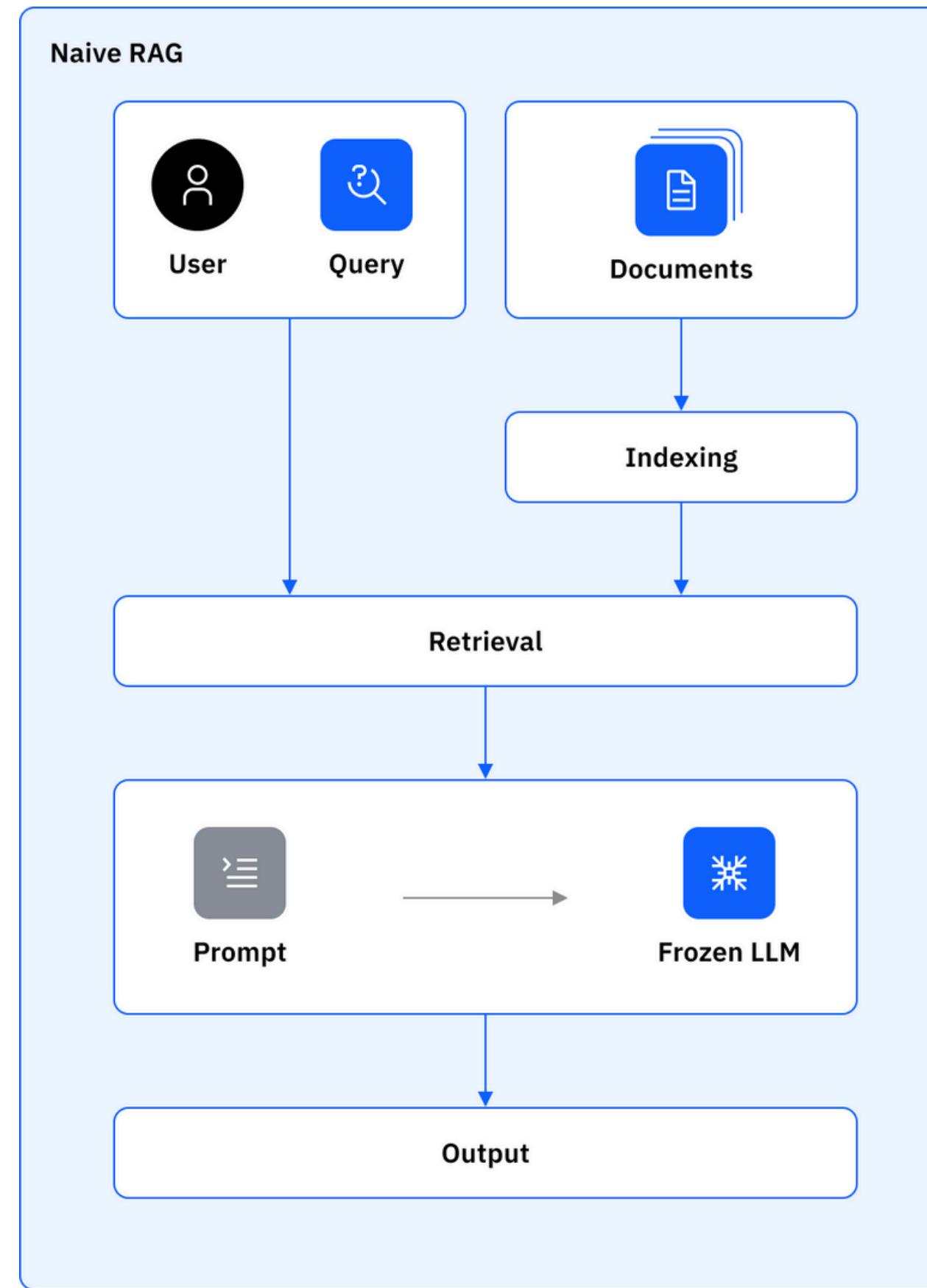
Fig illustrates the three-step process of how naive RAG works.

RAG paradigm

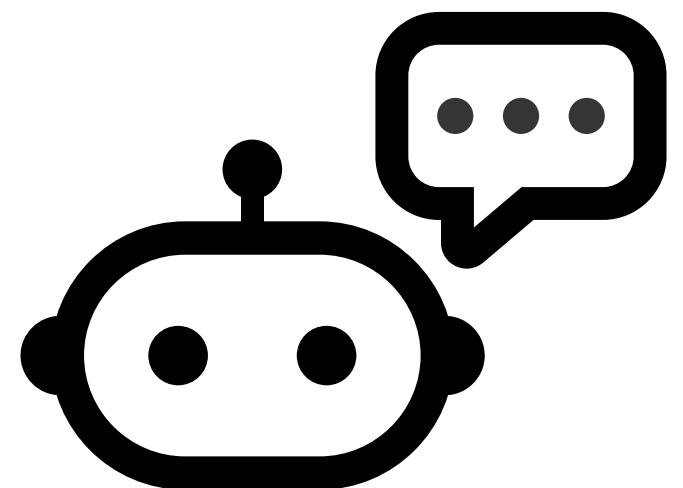
A. Naive RAG

three-step process for retrieval and content generation.

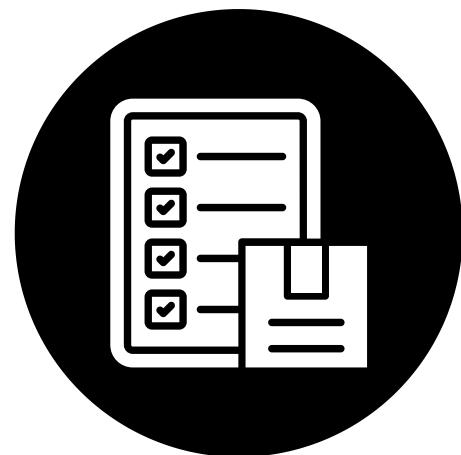
- 1. Query Encoding:** Transformed user's query into high-Dimension **Vectors**, capturing semantic meaning of query
- 2. Document Retrieval:** A similarity search is performed, though vector database retrieving N-documents
- 3. Response Generation:** served context and user's query to LLM



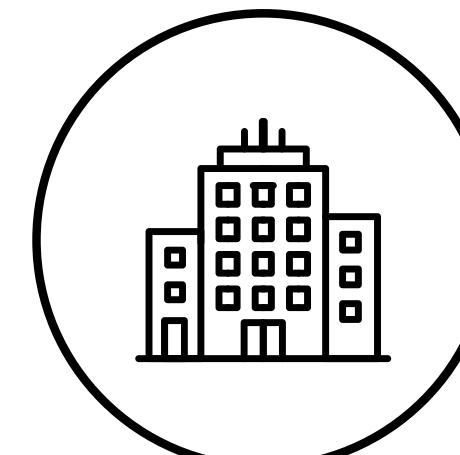
Applications of naive RAG



a. Customer support chatbots:
Handling frequently asked repetitive question-answering scenarios by using LLM responses



b. Summarization and information retrieval: Providing a basic level of summarization by using natural language processing techniques.

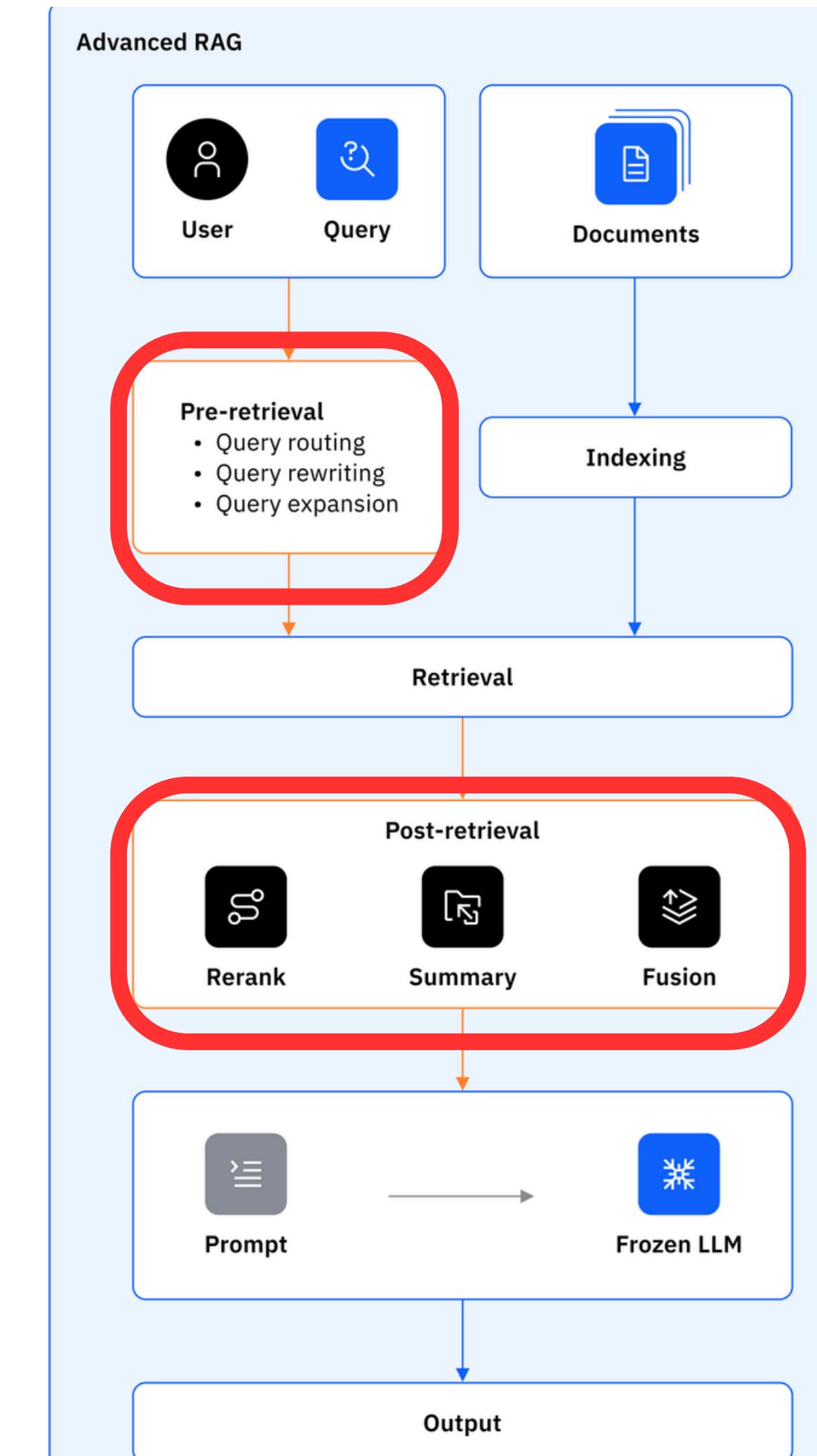


c. AI systems for enterprises:
Quickly retrieving relevant data from repositories to answer common queries.

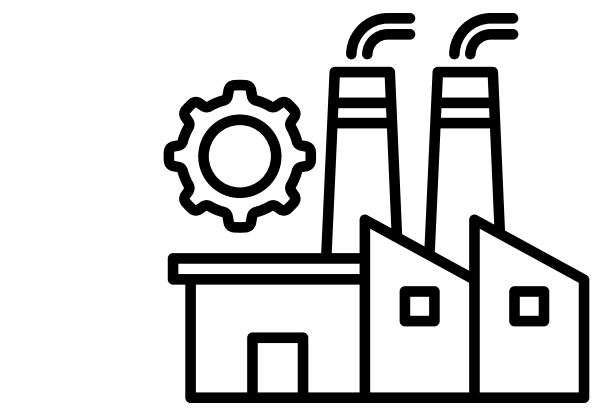
B. Advanced RAG

the power of **better retrieval and generation** by using sophisticated algorithms

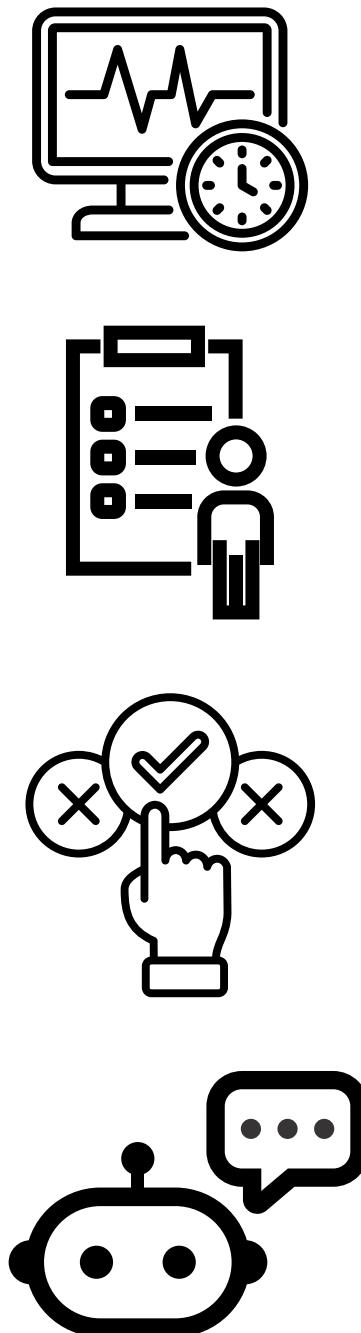
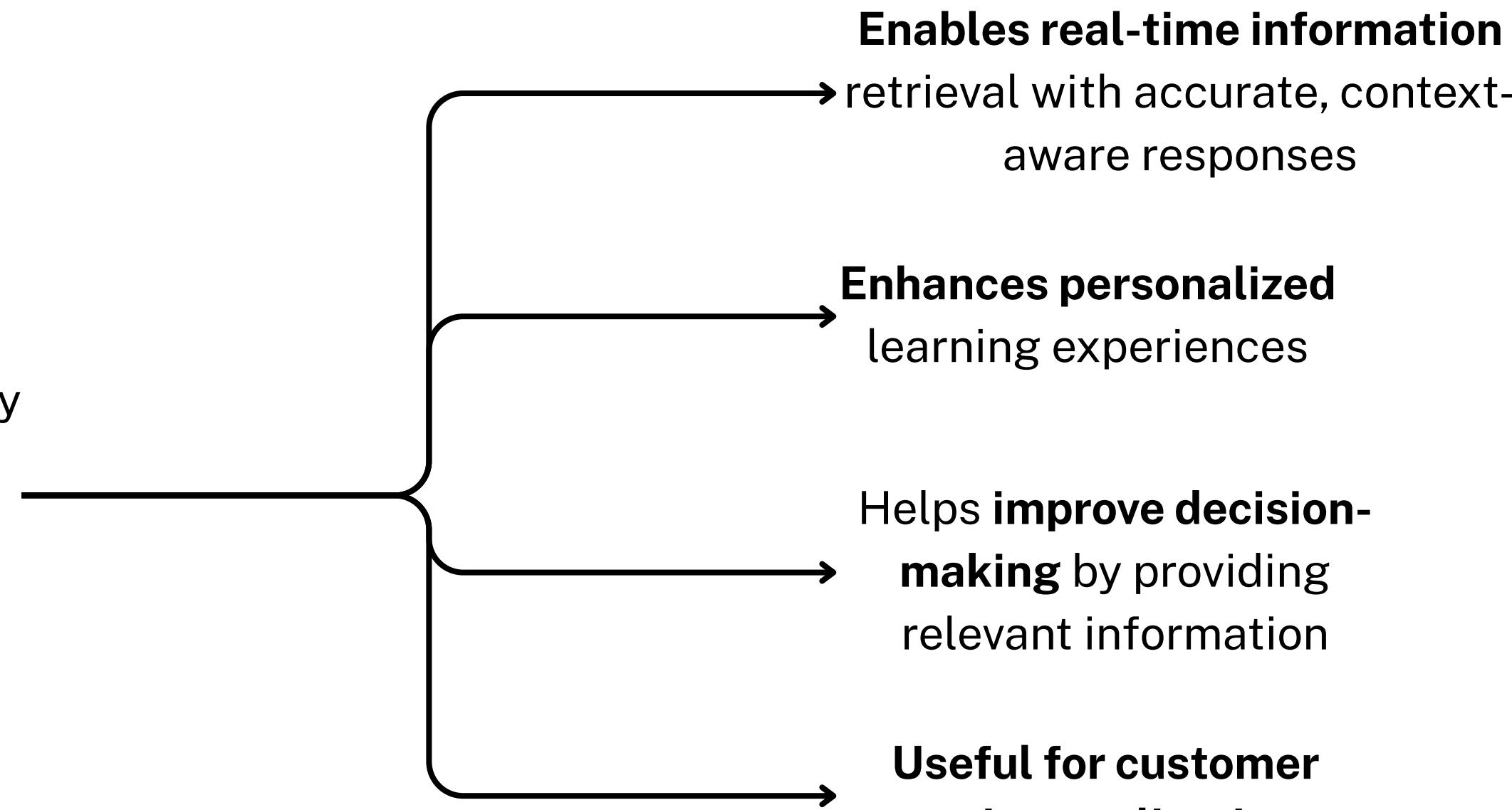
1. **Query Encoding**
2. **Document Retrieval**
3. **Reranking retrieved documents (+)**: The retriever **gives a final score based on context and in relation to the query** retrieving the documents.
4. **Contextual fusion for generation (+)**: each document is encoded differently, the decoder **fuses all encoded contexts to ensure that the generated responses have coherence with to the encoded query**.
5. **Response generation**
6. **Feedback loop (+)**: various techniques like active learning, **RL** and retriever-generator cotraining to continuously enhance its performance , **explicit feedback**.



Applications of Advanced RAG



Advanced RAG is highly versatile and can be **applied across many industries**



C.Modular RAG

By **disaggregating the act of RAG into modules**, one can better adapt, debug and optimize each component independently. Now let's see how modular RAG works in real action

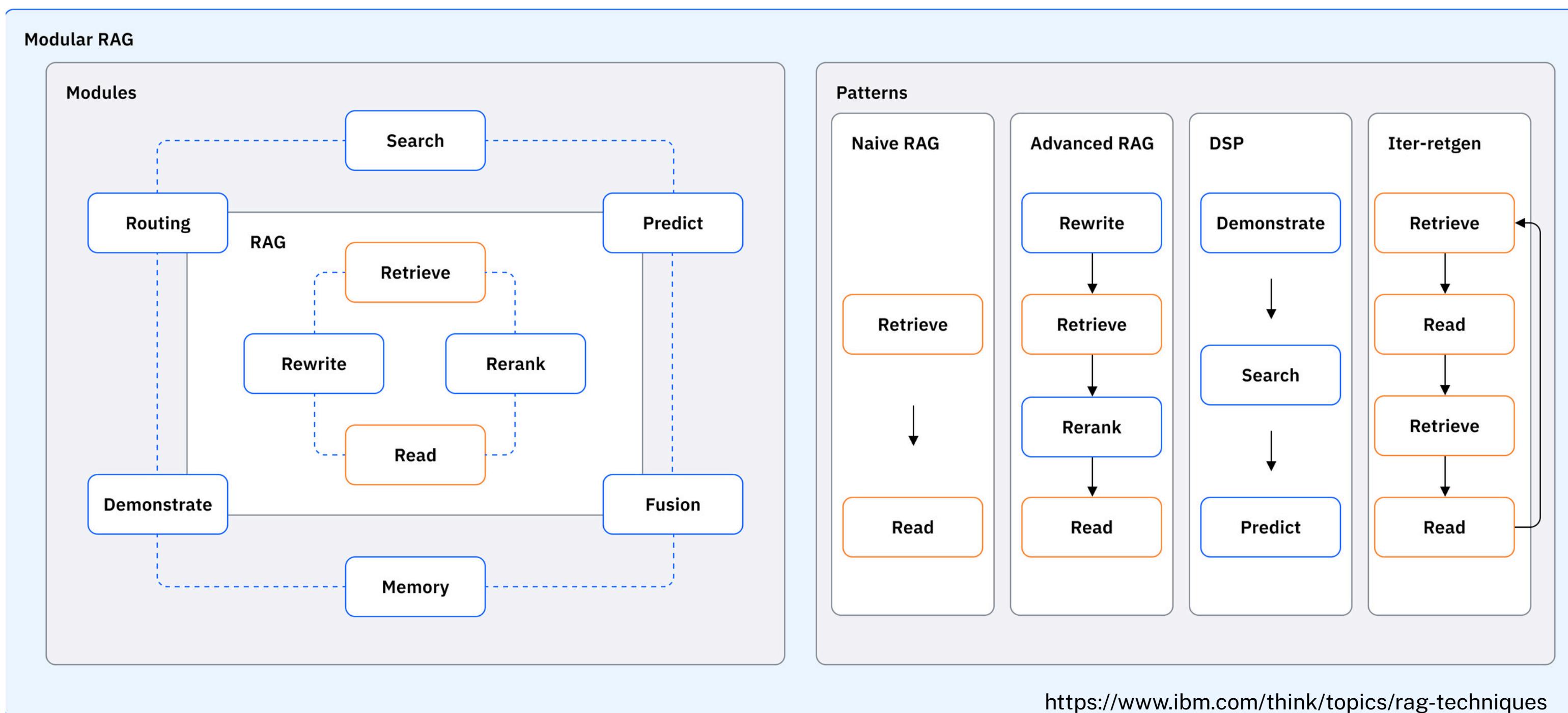


Fig illustrates the stepwise process of how modular RAG works.

Applications of Modular RAG

Modular RAG in Enterprises

1. **Customer Support Automation:** Deliver accurate, context-aware responses to customer queries.
2. **Knowledge Management:** Extract, summarize, and contextualize information from vast repositories.
3. **Content Creation:** Generate domain-specific marketing or technical content with high quality.
4. **Healthcare and Legal Industries:** Provide quick access to domain-specific knowledge, improving decision-making

Pros and cons of RAG paradigm

RAG Technique	Pros	Cons
Naive RAG	<ul style="list-style-type: none">• Straightforward to implement.• Quick to set up.	<ul style="list-style-type: none">• Lacks the flexibility and scalability required for intricate, real-world applications.
Advanced RAG	<ul style="list-style-type: none">• High Accuracy: Uses real-time, context-specific data to minimize errors.• Dynamic Adaptation: Incorporates user feedback (RLHF/Active Learning).• Domain Expertise: Integrates specialized databases.• Efficiency: Optimizes context windows by fetching only pertinent data.	<ul style="list-style-type: none">• Resource Heavy: High compute demands and significant resource needs for large knowledge bases.• Latency: Delays caused by both retrieval and generation steps.• Complexity: Hard to implement and maintain (requires fine-tuning multiple components).
Modular RAG	<ul style="list-style-type: none">• Flexibility & Customization: Components (retrievers, rankers, generators) can be swapped or fine-tuned independently.• Scalability: Modules can be distributed across various resources.• Maintainability: Easier debugging and updates without disrupting the whole system.• Cost-Effective: Optimizes retrieval and minimizes LLM usage.	<ul style="list-style-type: none">• Requires framework-specific knowledge like LangChain• Requires careful system design and orchestration• Complexity: Hard to implement and maintain (requires fine-tuning multiple components).

Search techniques

- 1. Keyword Search (Sparse Retrieval)**
- 2. Vector Search (Dense Retrieval)**
- 3. Hybrid Search**

Sparse Vector Search (Keyword Search)

Keyword Search operates on the basis of **Lexical Matching** between the query and the document.

BM25 is the OG of information retrieval. It ranks documents based on term frequency, inverse document frequency (**TF-IDF**), and length normalization.

IDF (Inverse Document Frequency): How rare is this term?

- “the” appears in 99% of documents → IDF = low (not useful)
- “JWT” appears in 2% of documents → IDF = high (very useful)

Length Normalization : Prevent long documents from winning just by being long

TF (Term Frequency): How often does it appear in this document?

$$\text{BM25}(t, d) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{f(t, d) \cdot (k_1 + 1)}{f(t, d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{\text{avgdl}})}$$

where:

- $f(t, d)$ is the frequency of t in d ,
- $|d|$ is the length of document d ,
- avgdl is the average document length in the collection,
- k_1 and b are hyperparameters that control the weighting.

Pros

- Finds exact keywords
- Great for technical jargon
- Understands term importance

Cons

- Slower than dense
- Misses semantic relationships
- Struggles with synonyms

Sparse Vector Search Use case

Best for: Specific identifiers, exact codes, or industry **jargon** where "close enough" is not good enough.



Scenario A: Law

User Query: "w.s.u. คอมพิวเตอร์ มาตรา 14"



Scenario B: E-commerce / Inventory

User Query: "MBP-M3MAX-32-1TB" (Specific SKU for a MacBook)

Dense Vector Search (Semantic Understanding)

Dense Vectors **capture Meaning** excellently. They handle ambiguous or conversational queries naturally.

These vectors are stored in a Vector Database. Search is performed by calculating the distance between the query vector and document vectors using metrics like:

- **Cosine Similarity:** Measures the angle between vectors (most popular for semantic similarity).

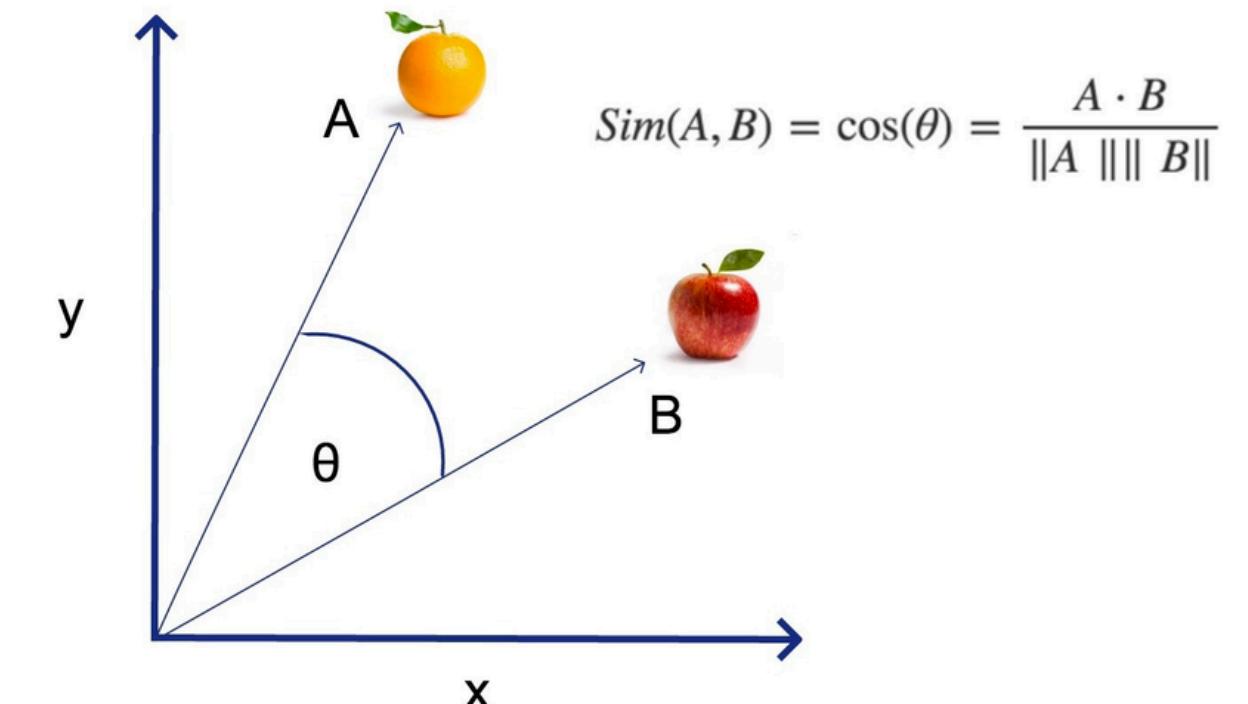
Pros:

- Fast
- Understands meaning

Cons:

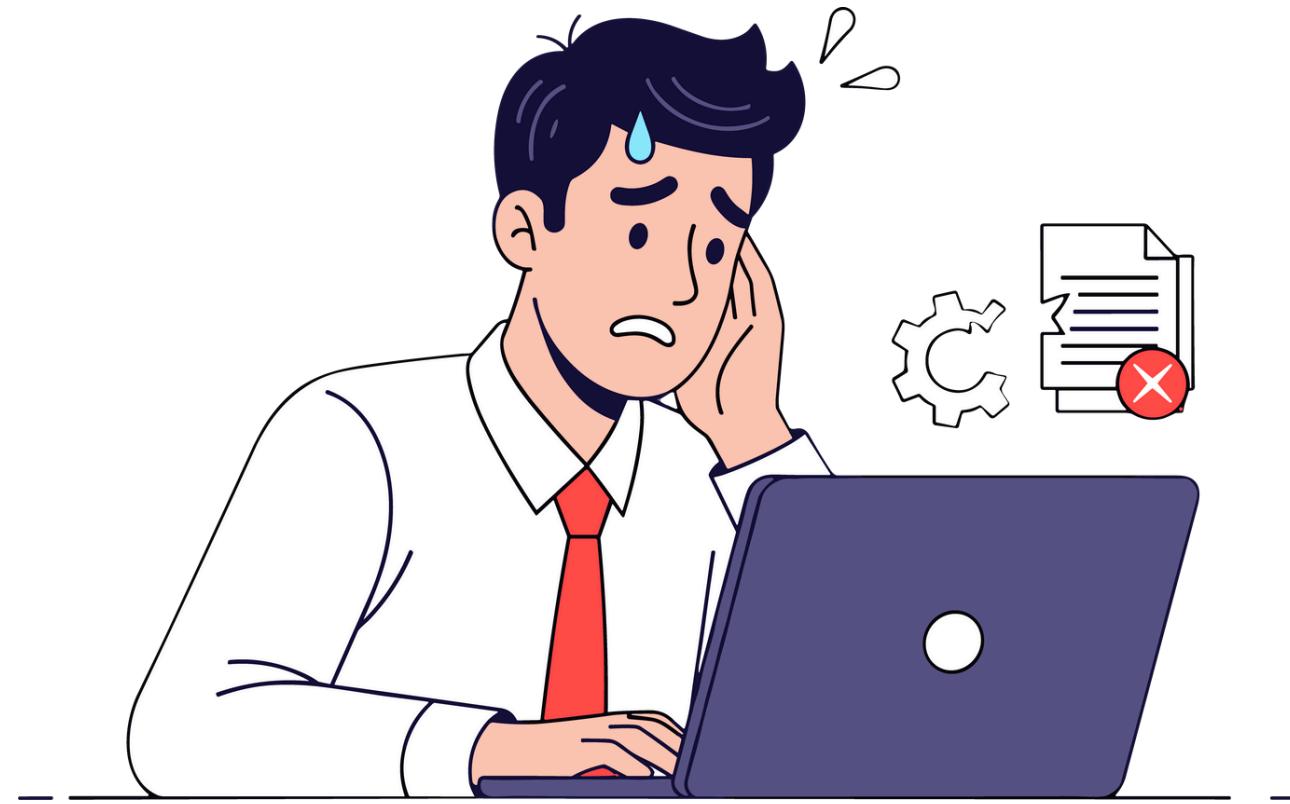
- Misses exact keyword matches
- Struggles with technical jargon (if the embedding model wasn't trained on it)

Cosine Similarity



Dense Vector Search Use cases

Best for: Natural language questions, describing a concept without knowing the right words, or finding synonyms.

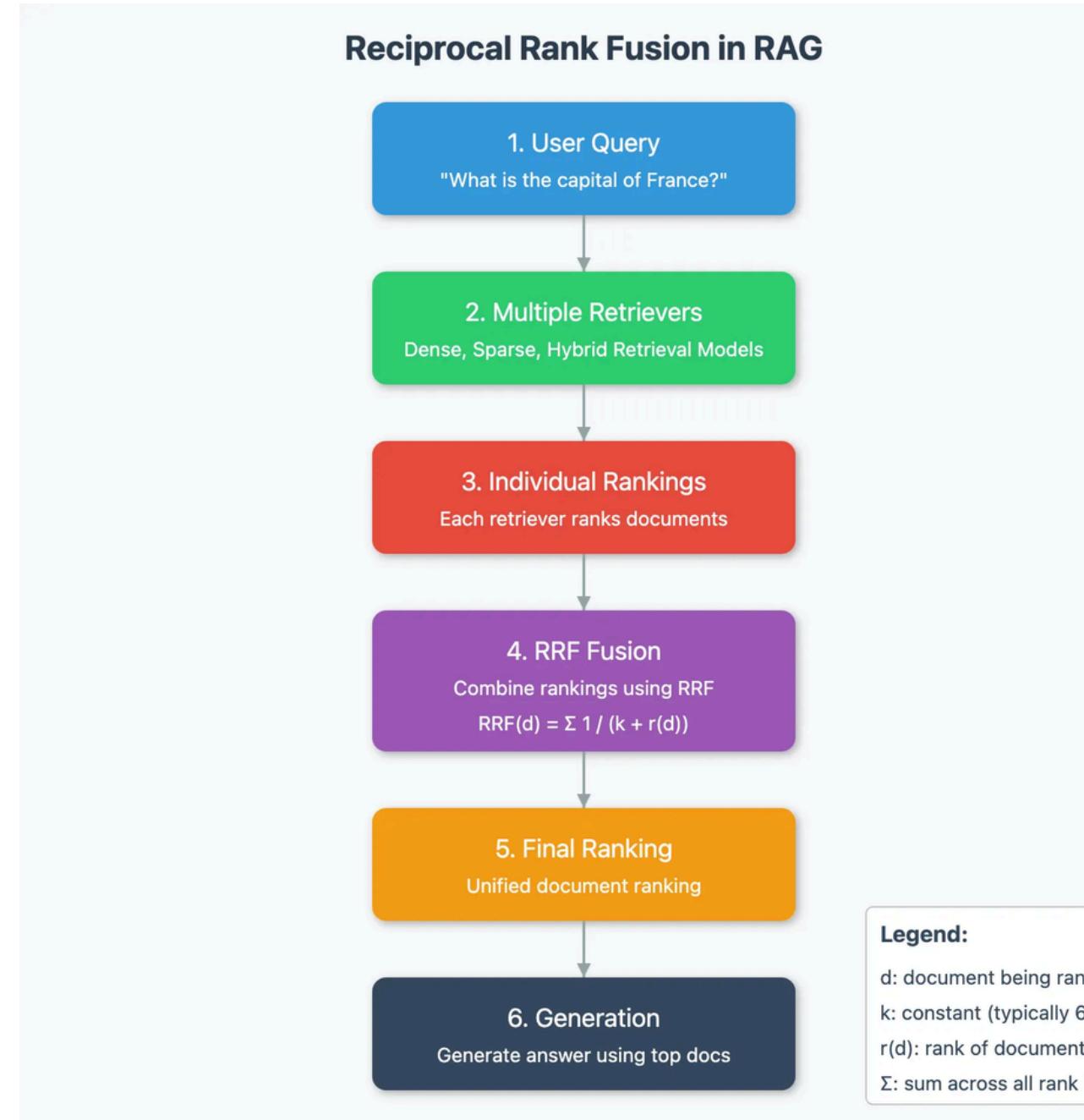


Scenario A: Customer Service (Intent Matching)
User Query: "How do I fix my internet connection?"

Scenario B: Discovery / Exploration
User Query: "Gifts for someone who loves outdoor activity"

Hybrid Search - RRF

The combination of the two (often using **RRF - Reciprocal Rank Fusion** to merge results).



Reciprocal Rank Fusion is a rank aggregation method that combines rankings from multiple sources into a single, unified ranking.

$$\text{RRF}(d) = \sum_{r \in R} \frac{1}{1 + r(d)}$$

Where:

- *d is a document*
- *R is the set of rankers (retrievers)*
- *k is a constant (typically 60)*
- *r(d) is the rank of document d in ranker r*

Hybrid Search - RRF Calculation example

Result from both Search

List A (Sparse): [Doc1, Doc2, Doc3]

List B (Dense): [Doc2, Doc4, Doc1]

Apply RRF formula

$$\text{Doc1} = \frac{1}{60+1} + \frac{1}{60+3} = 0.016 + 0.015 = 0.031 \text{ (2nd)}$$

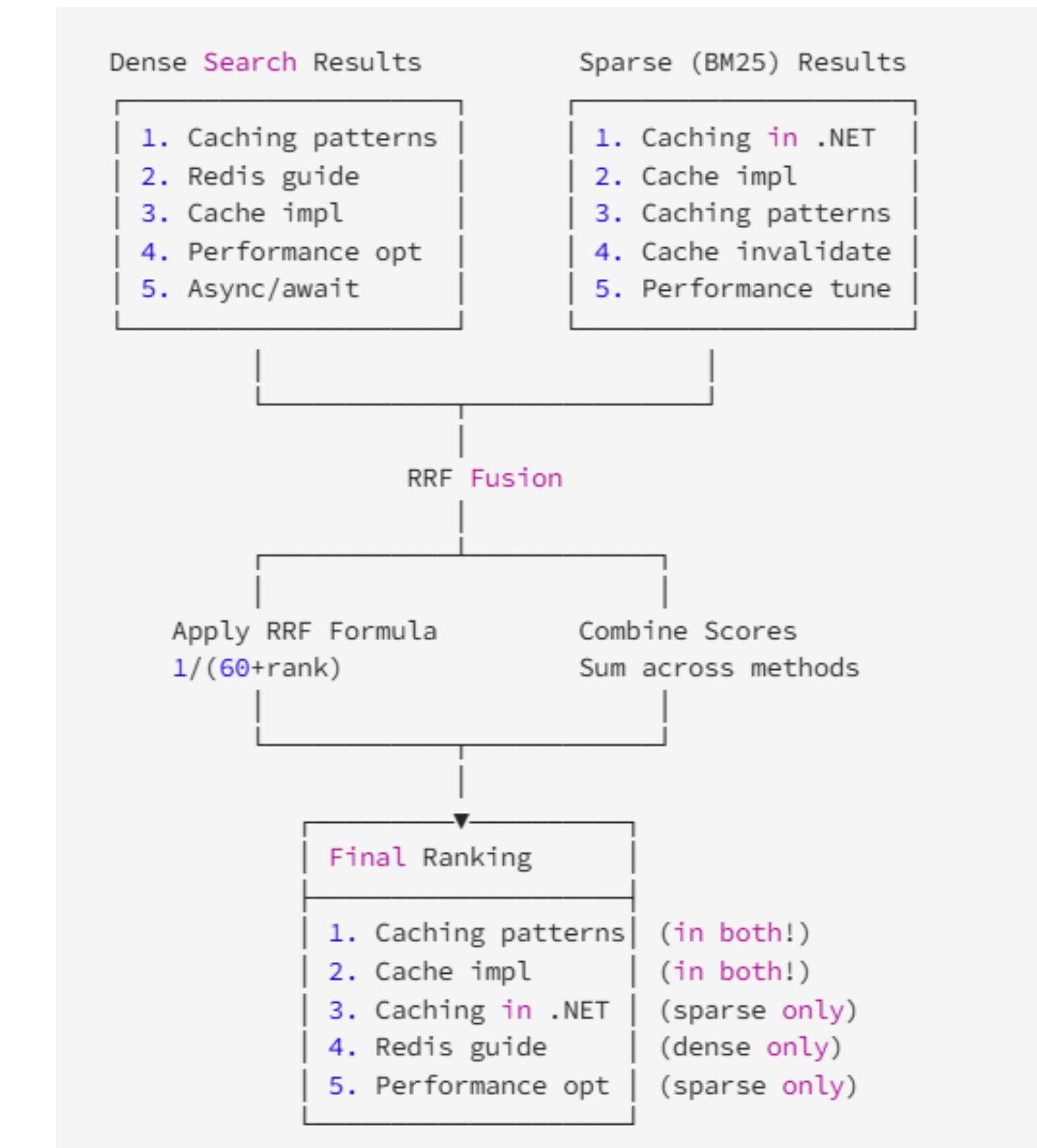
$$\text{Doc2} = \frac{1}{60+2} + \frac{1}{60+1} = 0.016 + 0.016 = 0.032 \text{ (1st)}$$

$$\text{Doc3} = \frac{1}{60+3} = 0.015 \text{ (4nd)}$$

$$\text{Doc4} = \frac{1}{60+2} = 0.016 \text{ (3rd)}$$

RRF final ranking

doc = [Doc2, Doc1, Doc4, Doc3]



Hybrid Search Use case

Best for: Complex queries that contain both specific constraints (names, places, codes) and general concepts.



Scenario A: Real Estate

User Query: "Quiet 3-bedroom home in Bangkok"



Scenario B: Shopping

User Query: "Natural fragrance without aluminum"

Search Techniques Performance Comparison

Technique	Latency	NDCG@10	MRR	Precision@10	Recall@10	Improvement	Best For
Dense Only	45ms	0.72	0.68	0.72	0.65	Baseline	Semantic queries, speed critical
Sparse (BM25)	120ms	0.58	0.55	0.58	0.52	-19% vs Dense	Keyword queries, technical terms
Hybrid (RRF)	165ms	0.85	0.82	0.85	0.78	+18% vs Dense	Mixed queries, best balance
Hybrid + HyDE	450ms	0.91	0.87	0.89	0.84	+26% vs Dense	Semantic + vocab mismatch

Hybrid is the answer: For 95% of use cases, hybrid RRF is the right choice

Retrieval Techniques

- 1. HyDE RAG**
- 2. Adaptive Retrieval**
- 3. Long RAG**
- 4. Graph RAG**

HyDE RAG (Hypothetical , retrieve+)

HyDE **generates synthetic documents** that hypothesize what a relevant answer might look like. This gives the system a richer context, even when the user's input is vague.

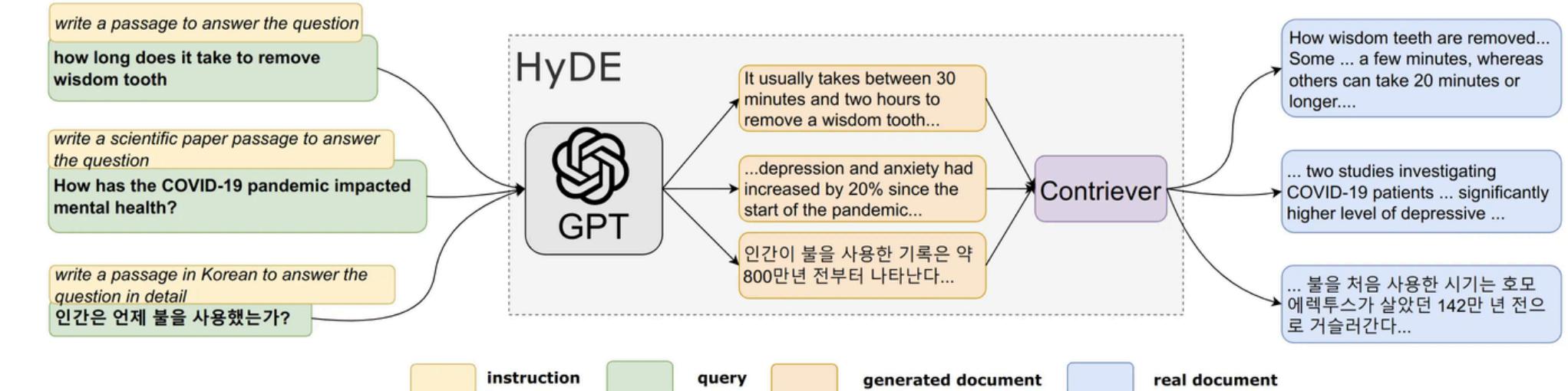
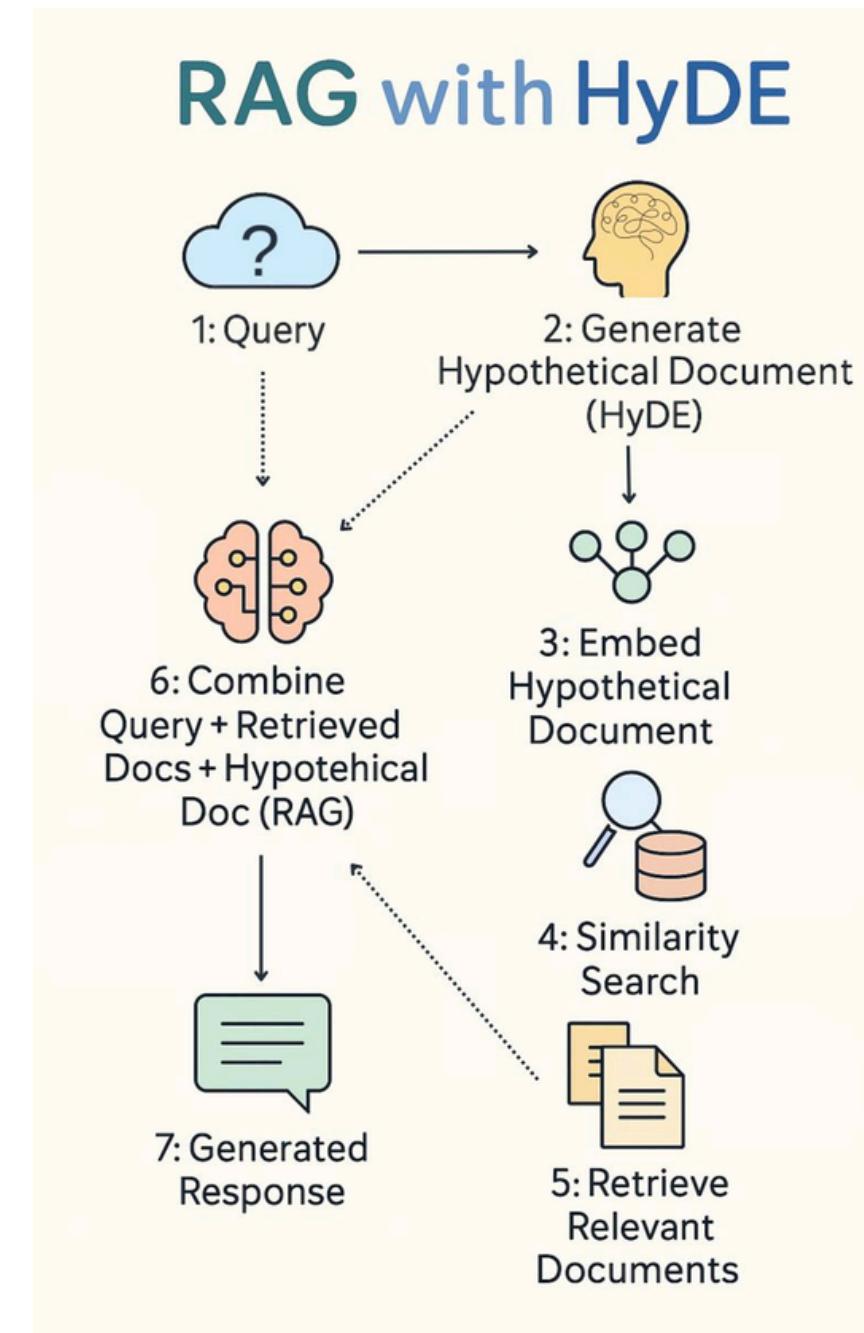


Figure 1: An illustration of the HyDE model. Documents snippets are shown. HyDE serves all types of queries without changing the underlying GPT-3 and Contriever/mContriever models.

<https://medium.aiplanet.com/advanced-rag-improving-retrieval-using-hypothetical-document-embeddings-hyde-1421a8ec075a>

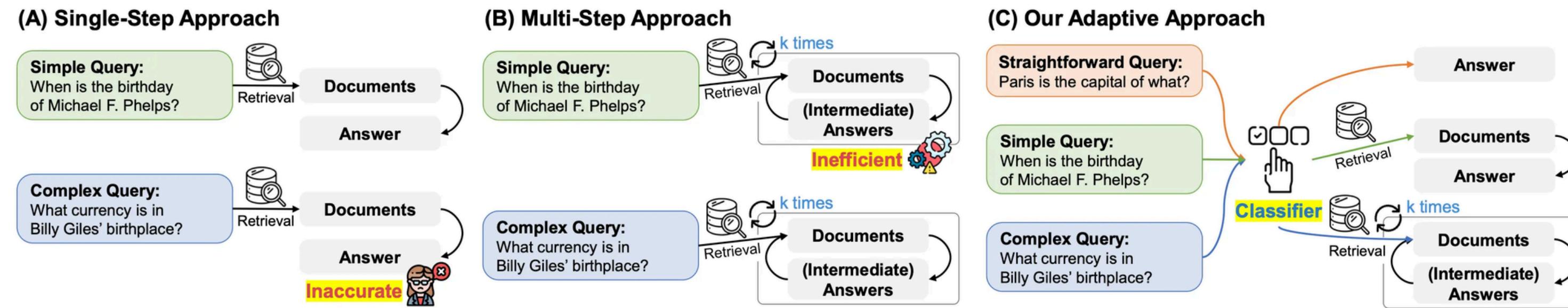
Best Use Case: HyDE is ideal for Technical Support or Research applications where accuracy is more important than speed, and where **user queries are often short or poorly phrased**.

Challenges

- **Risk of Hallucination** - from Inaccurate synthetic documents
- high-quality hypothetical documents **requires advanced models and infrastructure**

Adaptive Retrieval (routing,)

Adaptive RAG decides **when and how to retrieve based on the complexity of the question**. Unlike standard RAG, which blindly retrieves documents for every query.



Pros:

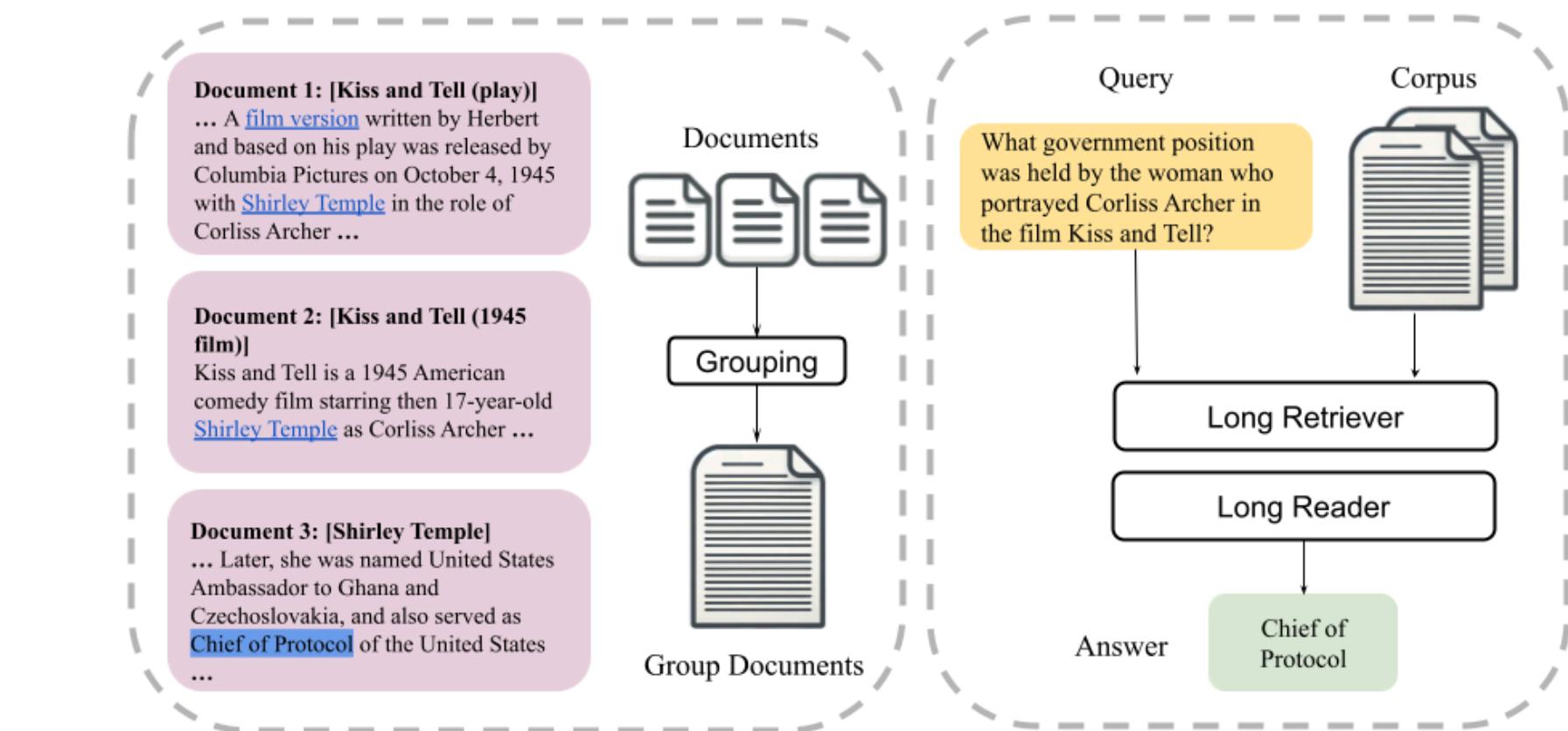
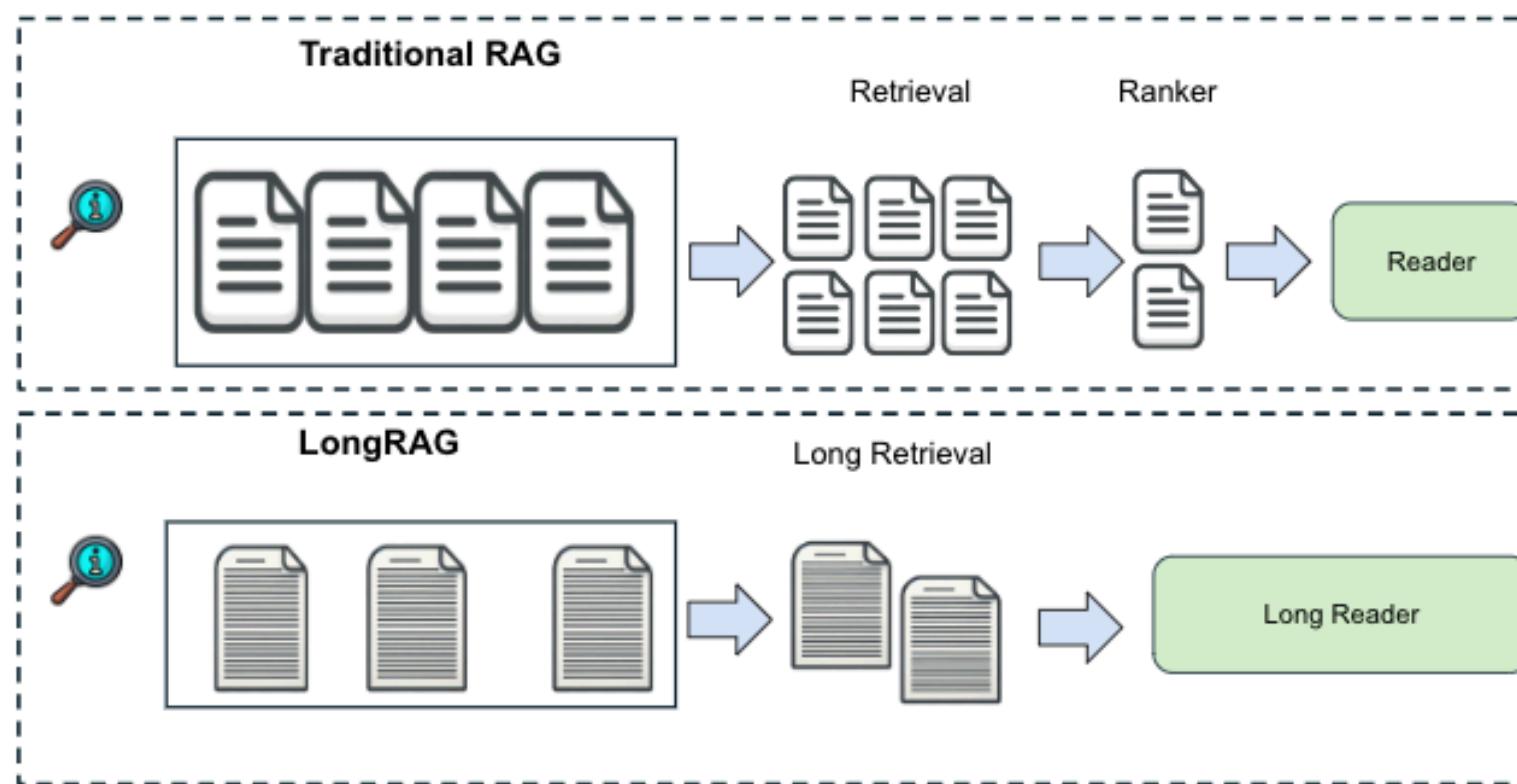
- Cost Efficiency
- Optimized Latency

Cons:

- Classifier Bottleneck

Long RAG (context scaling)

LongRAG operates on **long retrieval units** (30x longer). **Retriever** has a much less workload, which significantly boosts the recall score. LongRAG fully exploits the ability of **long-context language models (reader)** to achieve strong performance.



Pros:

- Improvement in Retrieval (Higher Recall)
- Reduced Information Fragmentation

Cons:

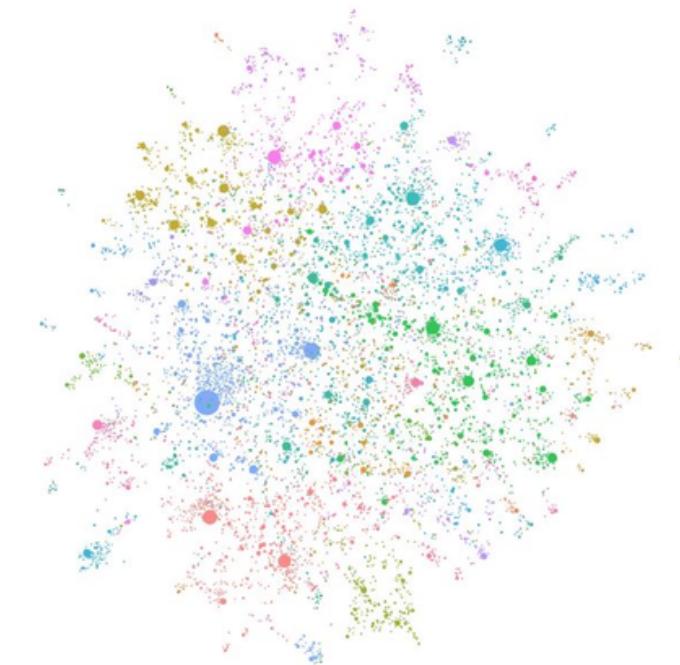
- Higher Latency (Slower Speed)
- LLM api Cost

Graph RAG (relation)

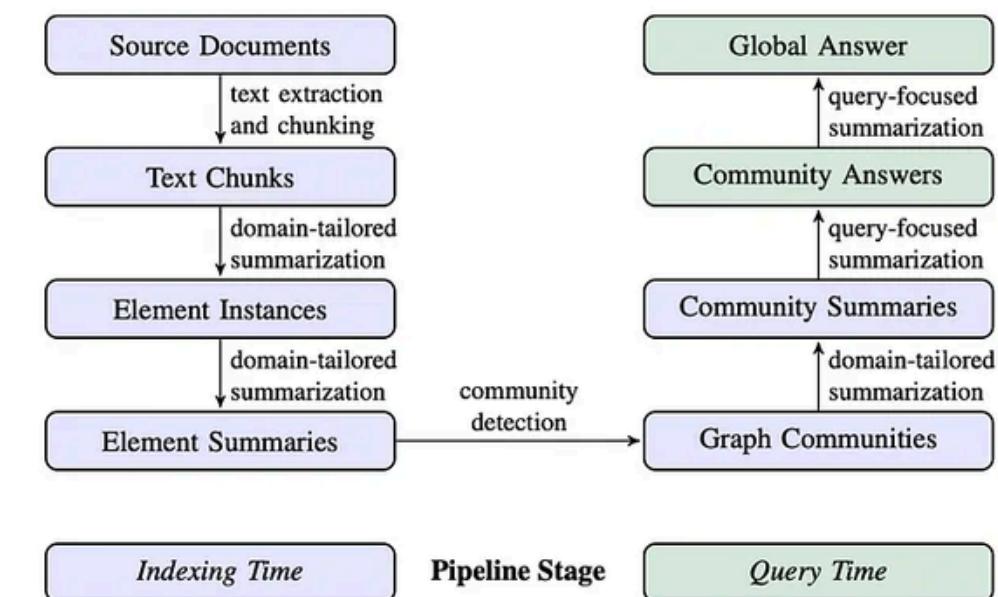
GraphRAG enhances RAG by incorporating **knowledge graphs (KGs)**. Knowledge graphs are data structures that **store and link related or unrelated data** based on their **relationships**.

Phase 1: Indexing

- **Text Unit Segmentation:** Dividing the raw input corpus into smaller, manageable pieces called **Chunks** (Text Units) to ensure detailed analysis.
- **Entity & Relationship Extraction:** Using an LLM to identify and extract **Entities** (people, places, organizations) and their Relationships from the text units to build the initial **knowledge graph**.
- **Hierarchical Clustering:** Utilizing techniques like the **Leiden Algorithm** to group related entities into "**Communities**" at different hierarchical levels.
- **Community Summary Generation:** Generating **summaries for each community** to provide a high-level overview of the information within that specific cluster.



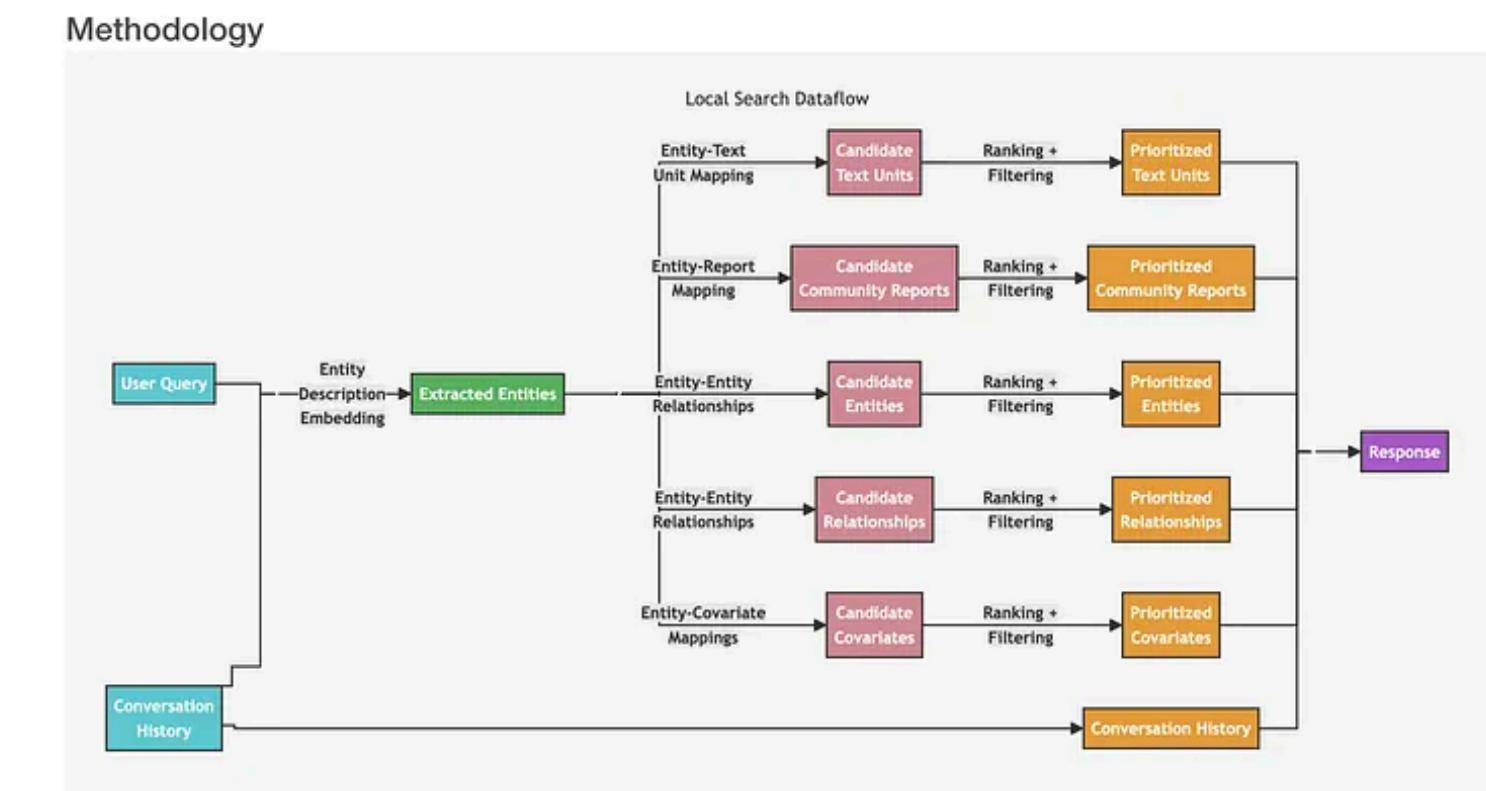
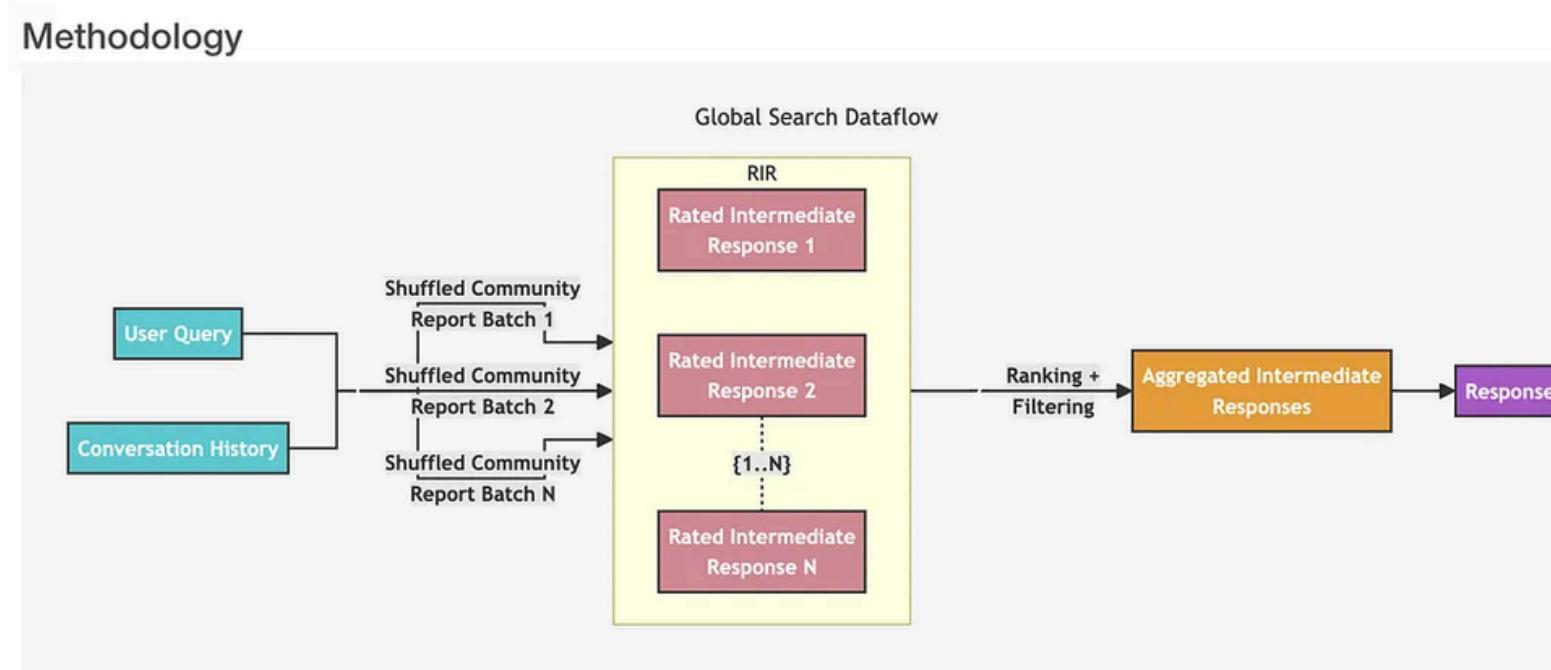
An LLM-generated knowledge graph built using GPT-4 Turbo.



Graph RAG (relation)

Phase 2: Querying

GraphRAG has **two** different querying workflows tailored for different queries.



Global Search for reasoning about holistic questions related to the whole data corpus by leveraging **the community summaries**.

Pros

- **Comprehensiveness:** Provides more complete answers by **identifying hidden relationships** that standard RAG might miss.
- **Diversity:** Offers a wider range of perspectives and insights by analyzing different "**Communities**" of data.
- **Contextual Understanding:** Better at **handling complex**, multi-hop reasoning than traditional Vector-only retrieval methods.

Local Search for reasoning about specific **entities** by fanning out to their **neighbors** and **associated concepts**.

Cons

- **Cost**
- **Computational Power**

Verification Techniques

- 1. Verification-Technique**
- 2. Corrective RAG**

Corrective RAG (CRAG)

Systems like CRAG trigger web searches to correct outdated retrievals

Pros

- Improved Performance and Real-Time Knowledge Integration
- Efficiency in Knowledge Utilization

Cons

- Detection and Correction of Wrong Knowledge
- Necessity of Fine-Tuning
- Potential Bias from Web Searches

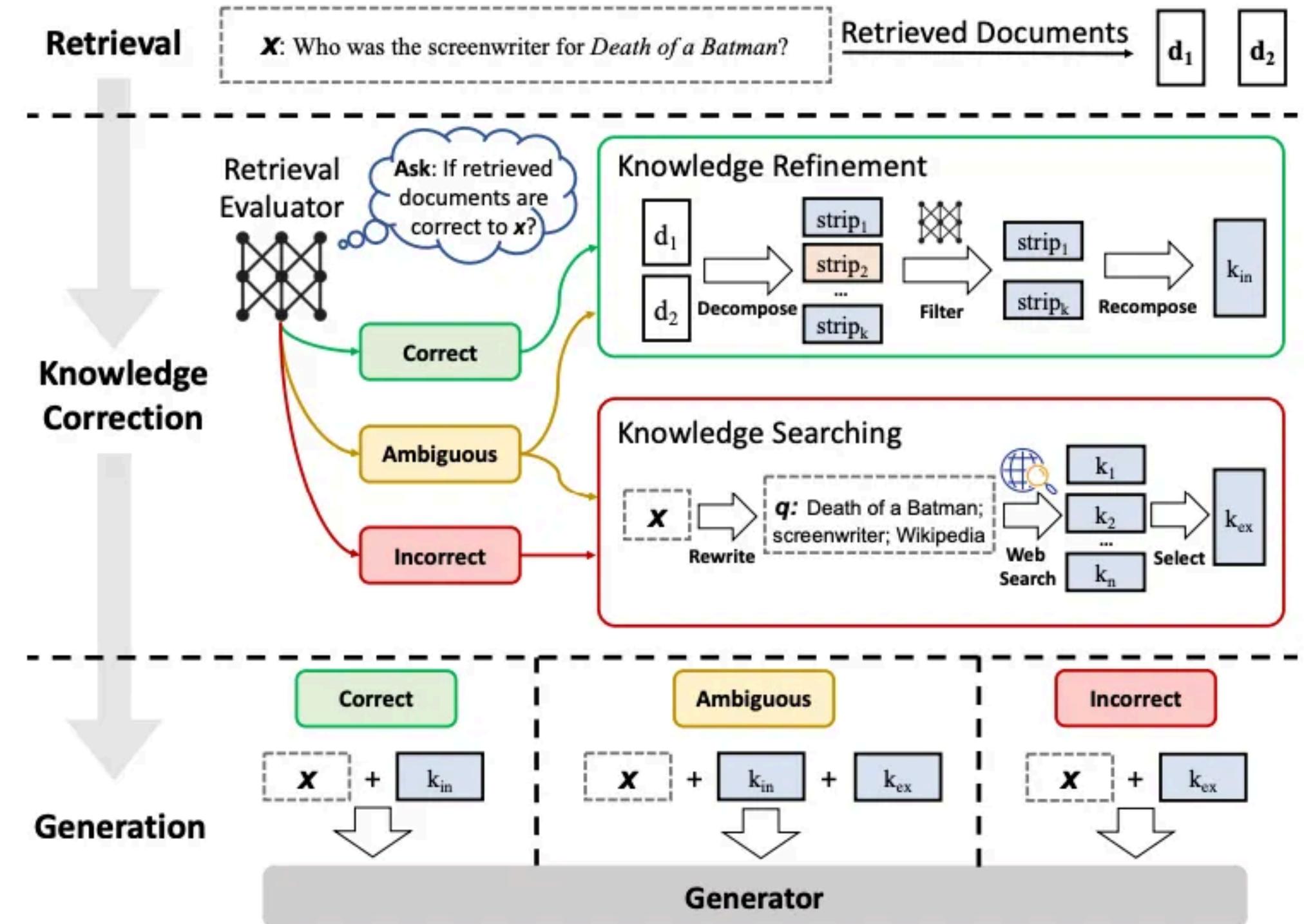


Fig illustrates Step-by-Step Process of CRAG

Self -RAG

Critiques its own retrievals using **reflection tokens**

Retrieval-Augmented Generation (RAG)

Prompt How did US states get their names?

Step 1: Retrieve K documents

1 Of the fifty states, eleven are named after an individual person.

2 Popular names by states. In Texas, Emma is a popular baby name.

3 California was named after a fictional island in a Spanish book.

Step 2: Prompt LM with K docs and generate

Prompt How did US states get their names? + 1 2 3

↓
LM → US states got their names from a variety of sources. Eleven states are named after an individual person (e.g., California was named after Christopher Columbus). Some states including Texas and Utah, are named after American tribe. No information in passages
Contradictory No information in passages

Prompt: Write an essay of your best summer vacation

1 2 3 → LM → My best...

Ours: Self-reflective Retrieval-Augmented Generation (Self-RAG)

Prompt How did US states get their names?

Step 1: Retrieve on demand

→ US states got their names from a variety of sources. Retrieve

Step 2: Generate segment in parallel

Prompt + 1



Relevant
11 of 50 state names
come from persons.
Supported

Prompt + 2



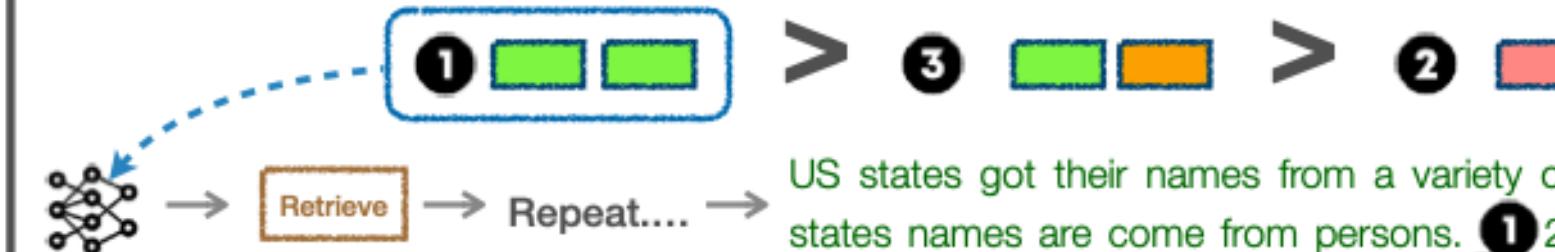
Irrelevant
Texas is named
after a Native American tribe.

Prompt + 3



Relevant
California's name has its
origins in a 16th-century novel
Las Sergas de Esplandián.
Partially

Step 3: Critique outputs and select best segment



→ Retrieve → Repeat... → US states got their names from a variety of sources. 11 of 50 states names are come from persons. 1 26 states are named after Native Americans, including Utah. 4

Prompt: Write an essay of your best summer vacation

↓
LM → No Retrieval → My best summer vacation is when my family and I embarked on a road trip along ...

Self -RAG

Pros

- **Reduced hallucination:** Answers are constrained to information supported by the retrieved evidence only.
- **High accuracy:** The system selects the best information from multiple sources.
- **Transparency:** It is possible to inspect why the AI chose a particular answer through critique or evaluation scores.

Cons

- **High resource usage:** Requires multiple rounds of generation and evaluation, increasing token consumption and response time.
- **System complexity:** Requires fine-tuning models to produce reflection tokens or the use of highly complex prompts.
- **Latency:** Slower than standard RAG due to multiple reasoning and evaluation stages.

Agentic RAG

Agentic RAG is the use of **AI agents** to facilitate **RAG**

Resoning Loop: using framework and dynamic workflow:

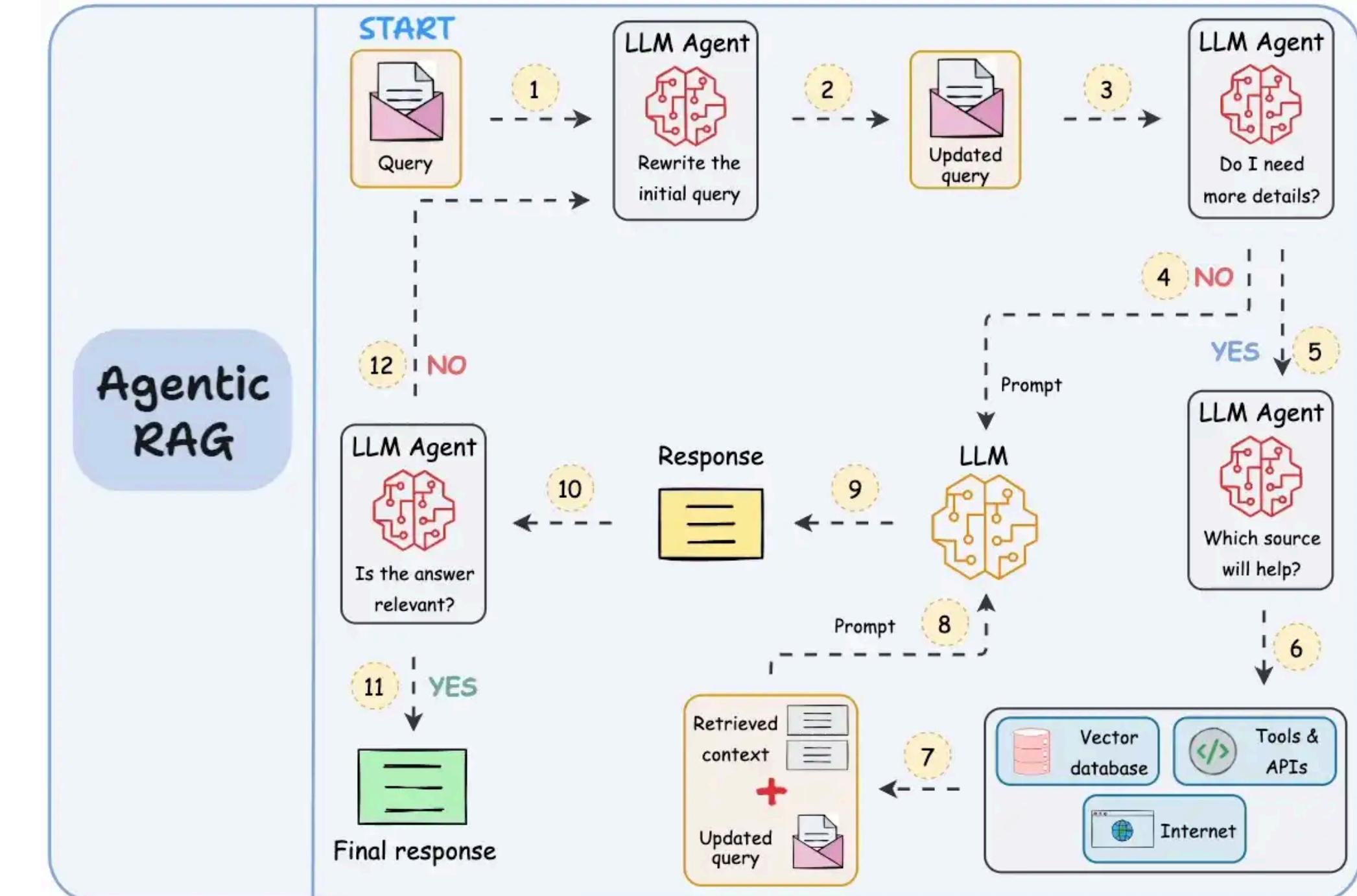
1. Query Decomposition & Planinng:

Agent analyze the user's prompt.
Break to sub-tasks

2. Tool Selection: Agent decide which tools to use

3. Iterative Retrieval: Agent execute a search looks at results **YES NO** If NO reformulates the query tries diff tools until YES to next step

4. Self-Correction & Validation



<https://blog.dailydoseofds.com/p/rag-vs-agentic-rag>

Agentic RAG

Pros

- **Excellent at solving complex problems:** Handles tasks requiring multiple data sources effectively (Multi-hop reasoning).
- **Intelligent data retrieval:** Smart enough to select the most appropriate method for pulling information (Similar to Adaptive RAG but more flexible).
- **Reduces Hallucinations:** Effectively minimizes false information through built-in **Verification steps.**

Cons

- **Latency (Slow):** Slower response times because the system must think and iterate through multiple loops.
- **Cost:** Higher expense due to increased **Token** usage from the reasoning process.
- **Unpredictable:** The Agent can sometimes "lose its way" or get stuck in an Infinite Loop without finding a solution.

Use case/ Application

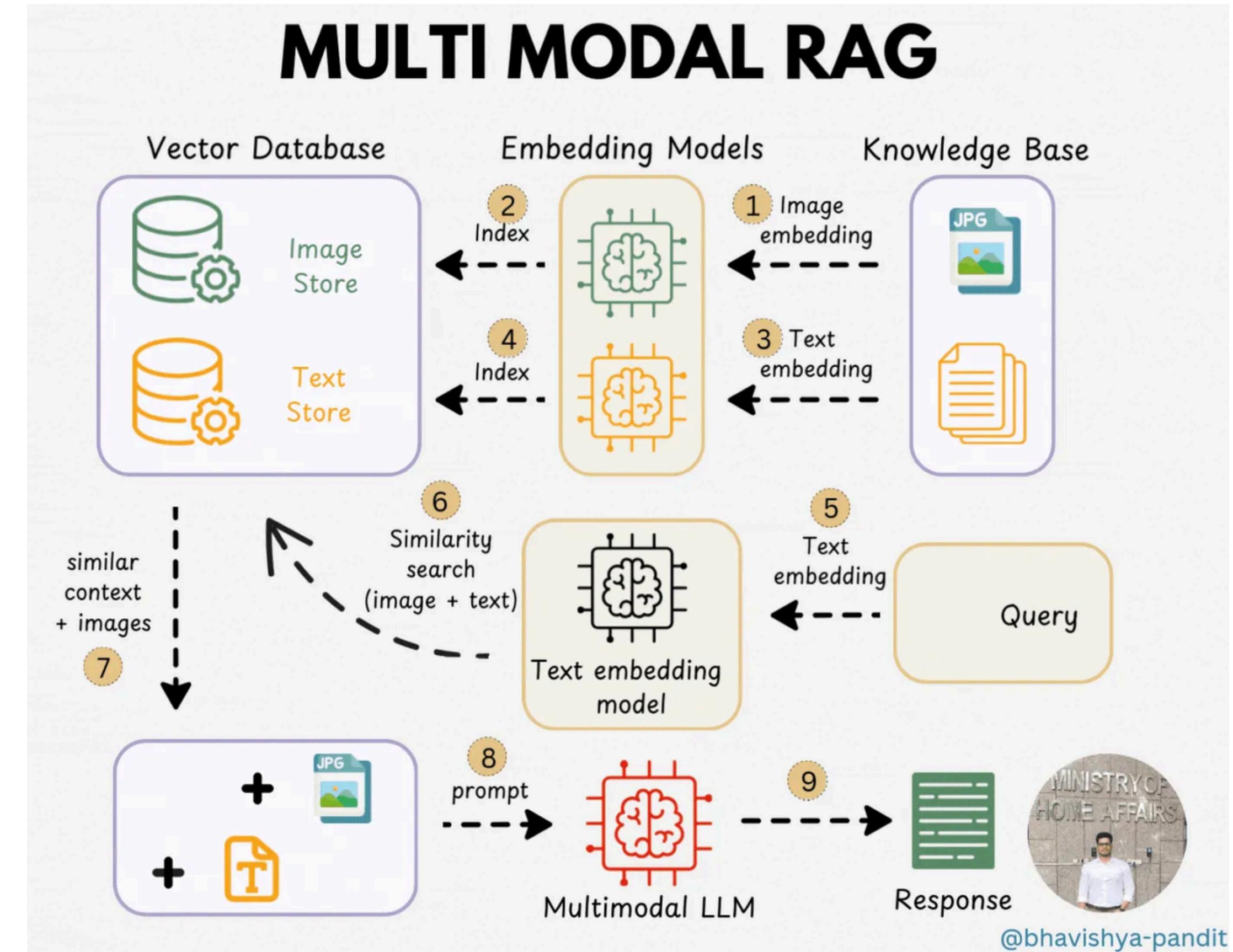
- **Real-time question-answering:** Enterprises can deploy RAG-powered chatbots and FAQs.
- **Automated support:** streamline customer support services.
- **Data management:** RAG systems make it easier to find information within proprietary data stores. Employees can quickly get the data.

Multimodal RAG

multimodal RAG can process and retrieve information from the multimodal data (text, images, tables, audio and video files) and generate contextually accurate and relevant responses.

Many Ways How multimodal RAG works:

1. **Captioning** (everything convert to text)
2. **Multimodal Embedding** (Shared Vector Space)
3. **Multi-vector Retriever** (Summary + Object Data)



Multimodal RAG

Pros

- **Understand Context:** Understand multimodal data to answer to user's query
- **User Experience:** image to text prompt
- **Decrease Hallucination**

Cons

- **Complexity:** manage index and data for retrieval and store multimodal data
- **Cost:** High Cost / Resource to process multimodal data
- **Storage:** store embedding of multimodal data and High usage memory

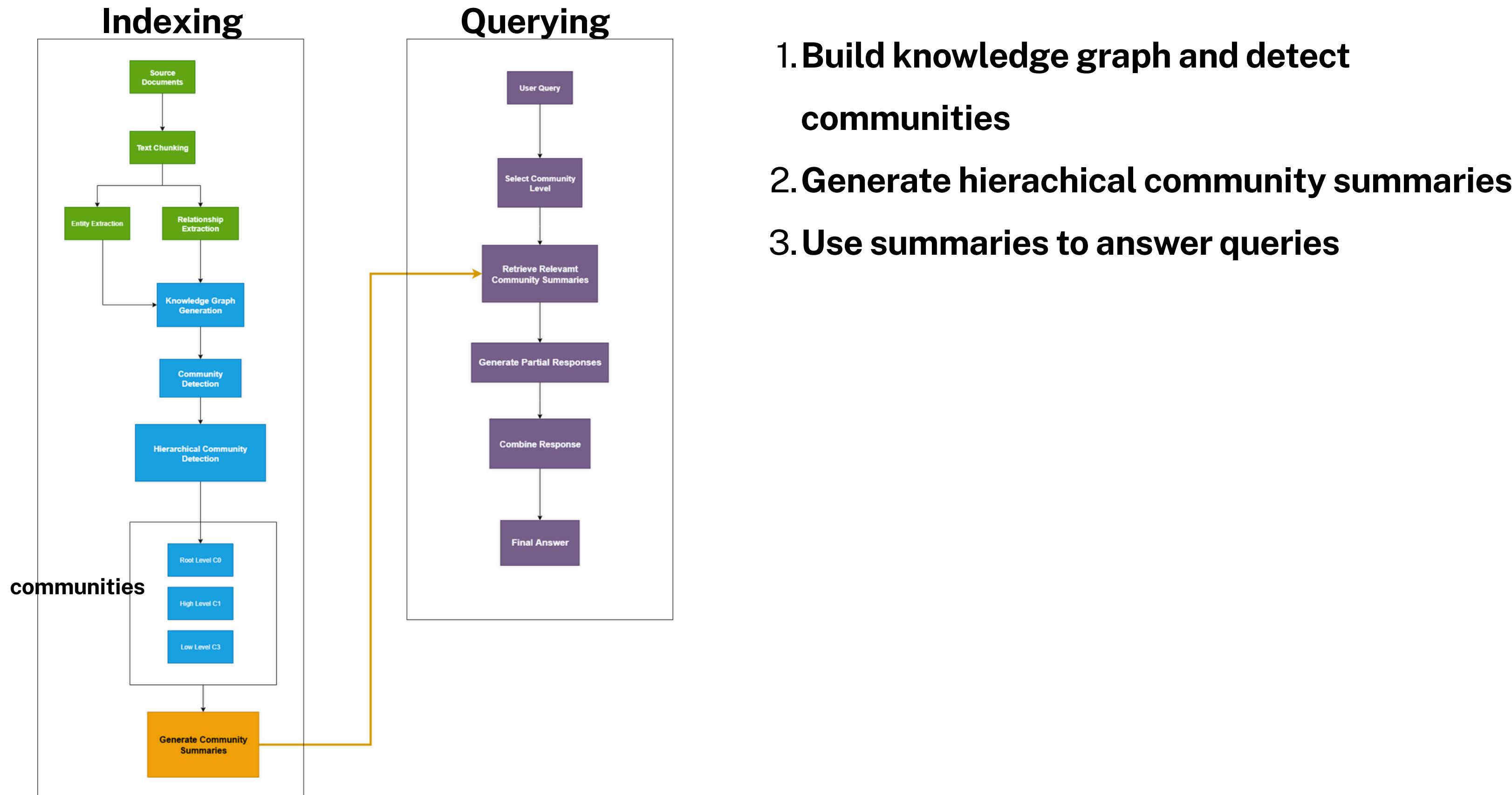
Use case/ Application

- **Industrial Application Study:** 1.) Healthcare and Medical Diagnostics, 2) Manufacturing and Industrial Maintenance, 3) Finance and Business Intelligence, 4) E-commerce and Retail, 5) Legal & Compliance etc.
- Multimodal RAG systems combine diverse data types to improve performance **open-domain question answering.**

Demo Graph RAG

Graph RAG - experiment

Experimented with example on <https://microsoft.github.io/graphrag>

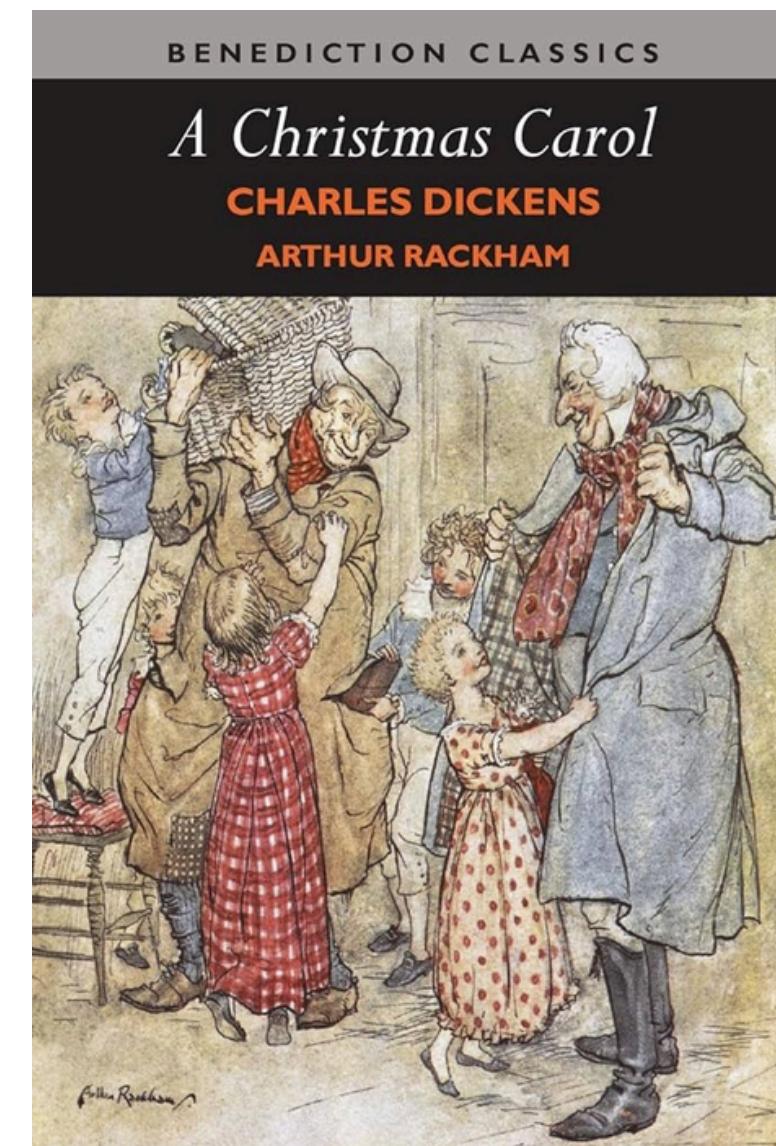


Graph RAG - experiment

Experimented with example on <https://microsoft.github.io/graphrag>

Input (.txt): A Christmas Carol by Charles Dickens, Arthur Rackham

The screenshot shows the GraphRAG documentation website. The top navigation bar includes links for Home, Indexing, Prompt Tuning, Query, Configuration, CLI, and Extras. The main content area features a "Getting Started" section with a warning about resource consumption, requirements (Python 3.10-3.12), installation options (Install from pypi or Use it from source), and an end-to-end example. At the bottom, there is a link to "Install GraphRAG".

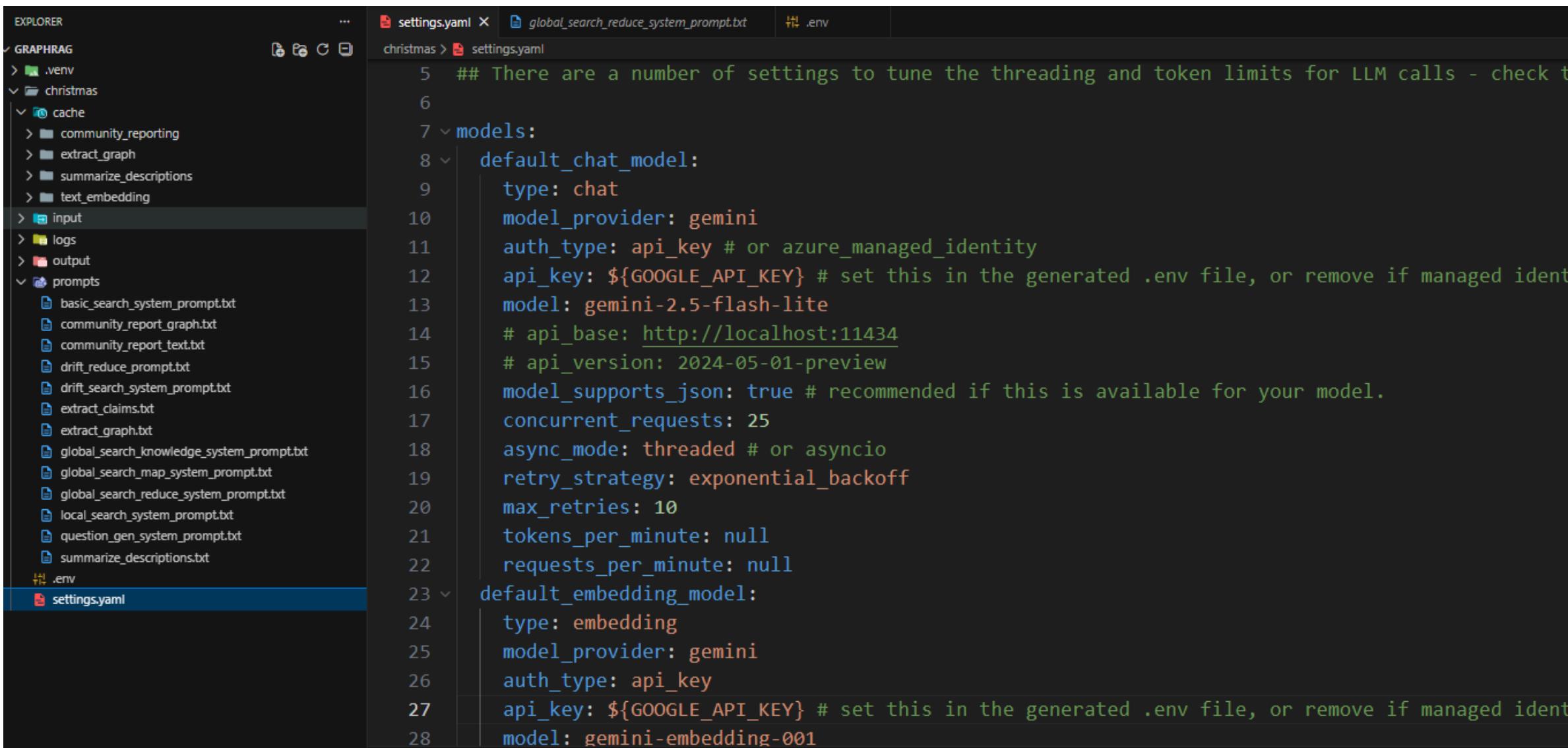


Graph RAG - experiment

Model Configuration

1. Chat model: gemini-2.5-flash-lite

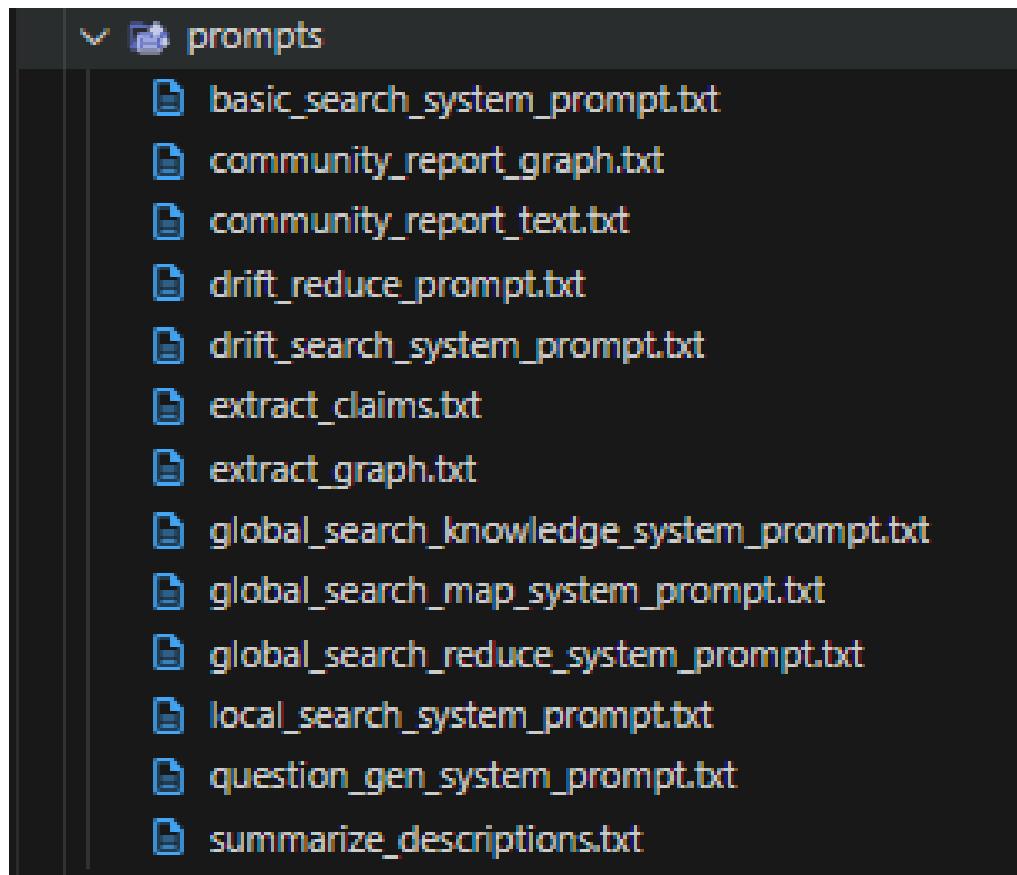
2. Embedding model: gemini-embedding-001



```
5 ## There are a number of settings to tune the threading and token limits for LLM calls - check t
6
7 models:
8   default_chat_model:
9     type: chat
10    model_provider: gemini
11    auth_type: api_key # or azure_managed_identity
12    api_key: ${GOOGLE_API_KEY} # set this in the generated .env file, or remove if managed ident
13    model: gemini-2.5-flash-lite
14    # api_base: http://localhost:11434
15    # api_version: 2024-05-01-preview
16    model_supports_json: true # recommended if this is available for your model.
17    concurrent_requests: 25
18    async_mode: threaded # or asyncio
19    retry_strategy: exponential_backoff
20    max_retries: 10
21    tokens_per_minute: null
22    requests_per_minute: null
23  default_embedding_model:
24    type: embedding
25    model_provider: gemini
26    auth_type: api_key
27    api_key: ${GOOGLE_API_KEY} # set this in the generated .env file, or remove if managed ident
28    model: gemini-embedding-001
```

Graph RAG - experiment

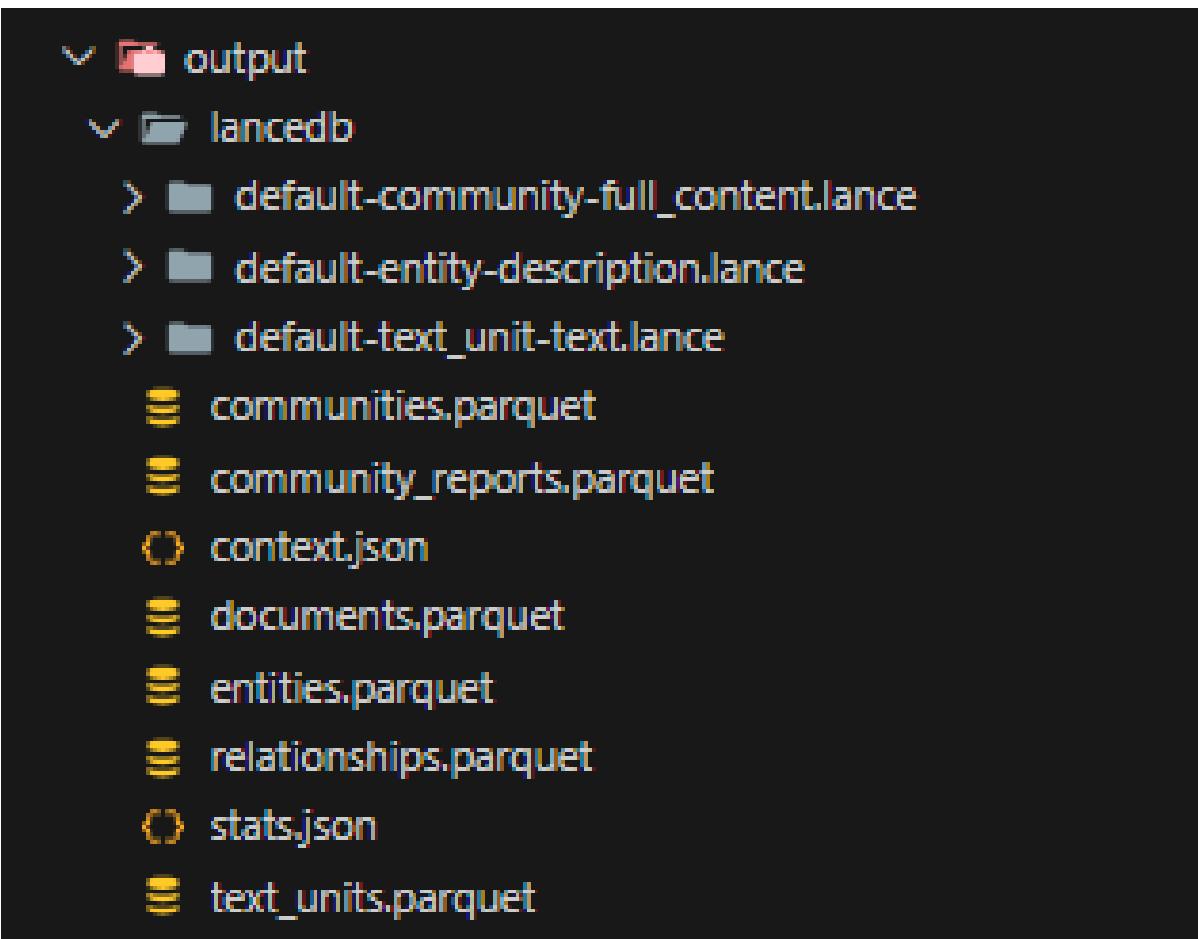
Prompts



Stage	Task	Prompt File Path
Indexing	Graph Extraction	prompts/extract_graph.txt
Indexing	Description Summarization	prompts/summarize_descriptions.txt
Indexing	Claim Extraction	prompts/extract_claims.txt
Indexing	Community Report (Graph)	prompts/community_report_graph.txt
Indexing	Community Report (Text)	prompts/community_report_text.txt
Search	Local Search System	prompts/local_search_system_prompt.txt
Search	Global Search (Map)	prompts/global_search_map_system_prompt.txt
Search	Global Search (Reduce)	prompts/global_search_reduce_system_prompt.txt
Search	Global Search (Knowledge)	prompts/global_search_knowledge_system_prompt.txt
Search	Drift Search System	prompts/drift_search_system_prompt.txt
Search	Basic Search System	prompts/basic_search_system_prompt.txt

Graph RAG - experiment

Indexing



.parquet files

	id	human_readable_id	title	type	description	text_unit_ids	frequency	degree	x	y
0	a543927c-b347-4cfa-a492-89b7fe79aa64	0	PROJECT GUTENBERG	ORGANIZATION	Project Gutenberg is a renowned initiative ded...	[336671e337e5f4539069473e8f8691b3ed696331aab...	4	8	0.0	0.0
1	5da89c14-9d36-4cf2-a86f-666e8973a640	1	CHARLES DICKENS	PERSON	Charles Dickens is the author of "A Christmas ...	[336671e337e5f4539069473e8f8691b3ed696331aab...	1	2	0.0	0.0
2	5190314f-bdde-4ba3-b2e8-b5866e398f85	2	ARTHUR RACKHAM	PERSON	Arthur Rackham is the illustrator of "A Christ...	[336671e337e5f4539069473e8f8691b3ed696331aab...	1	1	0.0	0.0
3	fd2d896a-9cc1-4453-8ebd-5efcb9762f8b	3	J. B. LIPPINCOTT COMPANY	ORGANIZATION	J. B. Lippincott Company is the original publi...	[336671e337e5f4539069473e8f8691b3ed696331aab...	1	6	0.0	0.0
4	7d999daf-f408-40ab-b31c-5a5465208277	4	SUZANNE SHELL	PERSON	Suzanne Shell was a producer of the Project Gu...	[336671e337e5f4539069473e8f8691b3ed696331aab...	1	1	0.0	0.0

entities.parquet

	id	human_readable_id	source	target	description	weight	combined_degree	text_unit_ids
0	069d621f-bcfa-4b38-a44b-165b45dfc017	0	PROJECT GUTENBERG	SUZANNE SHELL	Suzanne Shell produced the Project Gutenberg e...	7.0	9	[336671e337e5f4539069473e8f8691b3ed696331aab...
1	eab1a0f6-ab48-46a3-86f3-de2ca816126b	1	PROJECT GUTENBERG	JANET BLENKINSHIP	Janet Blenkinship produced the Project Gutenbe...	7.0	9	[336671e337e5f4539069473e8f8691b3ed696331aab...
2	7298b9d2-c51e-41ea-b4fa-235831d5215c	2	PROJECT GUTENBERG	ONLINE DISTRIBUTED PROOFREADING TEAM	The Online Distributed Proofreading Team assis...	7.0	9	[336671e337e5f4539069473e8f8691b3ed696331aab...
3	c4946d7d-b79b-4795-9719-1818eeae709b	3	PROJECT GUTENBERG	A CHRISTMAS CAROL (EBOOK #24022)	Project Gutenberg released the eBook #24022 of...	8.0	9	[336671e337e5f4539069473e8f8691b3ed696331aab...
4	83b010e1-a1cf-4185-b049-d7412917461b	4	PROJECT GUTENBERG	2007	The Project Gutenberg eBook #24022 of "A Chri...	8.0	9	[336671e337e5f4539069473e8f8691b3ed696331aab...

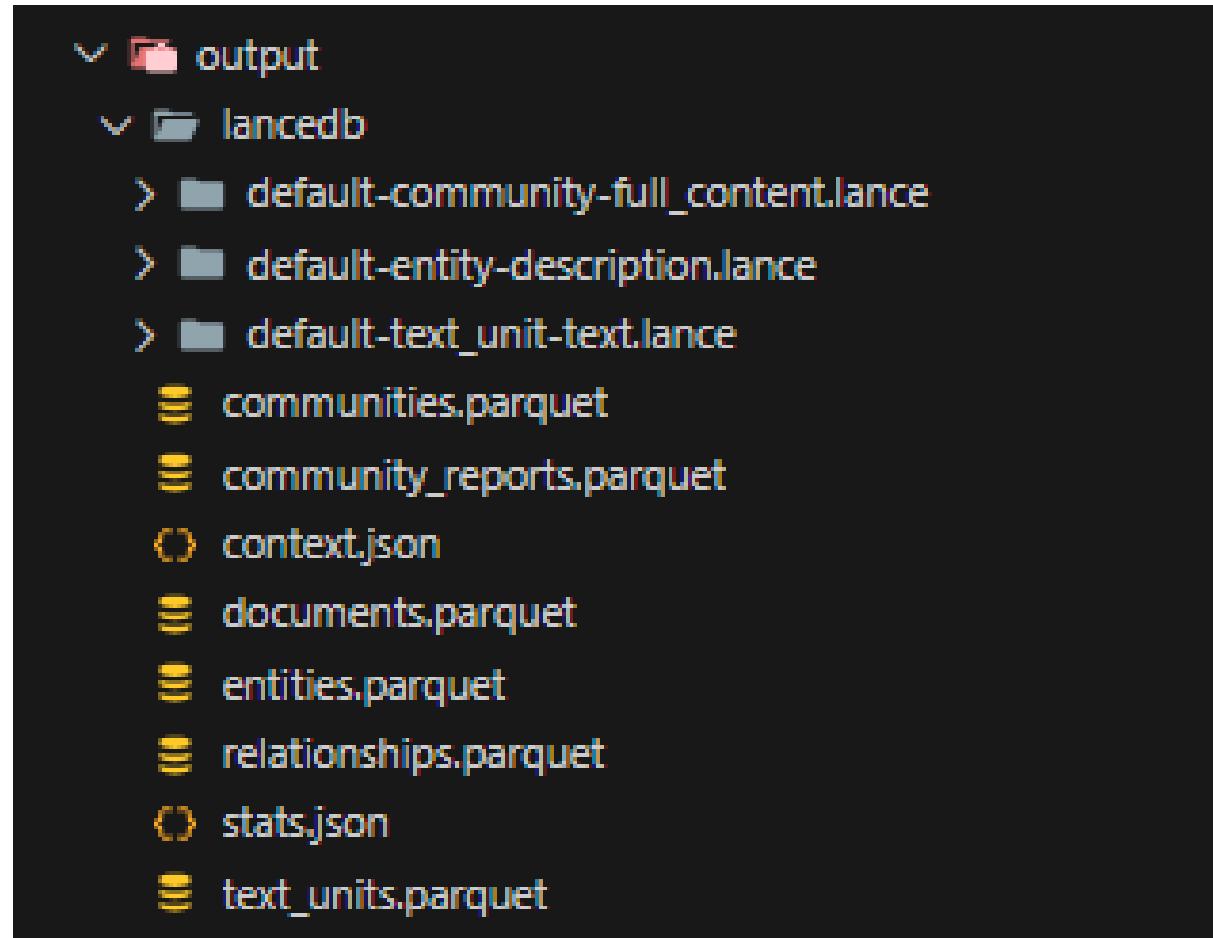
relationships.parquet

	id	human_readable_id	community	level	parent	children	title	summary	full_content	rank	rating_explanation	findings	full_content_json	period	size
0	ace58da9a9e44a5e982f357691bdab92	23	23	2	11	[]	Ebenezer Scrooge and His Transformative Christ...	This community centers around Ebenezer Scrooge...	# Ebenezer Scrooge and His Transformative Chri...	7.5	The impact severity rating is high due to the ...	{ "explanation": "Ebenezer Scrooge is the pivo..."}	{ "title": "Ebenezer Scrooge and His Tran..."}	2026-01-19	43
1	0ab48d33f54e481c8beca7384f7b6722	24	24	2	11	[]	Obscure Part of Town and Associated Shop	The community is centered around a specific lo...	# Obscure Part of Town and Associated Shop\n...	2.0	The impact severity rating is low due to the i...	{ "explanation": "The 'Obscure Part of Town' i..."}	{ "title": "Obscure Part of Town and Asso..."}	2026-01-19	2
2	e791a982c21d48b39a8d5c5ac2180a9e	25	25	2	20	[]	The Cratchit Family and Christmas Dinner	The community is centered around the Cratchit ...	# The Cratchit Family and Christmas Dinner\n...	3.0	The impact severity rating is low, reflecting ...	{ "explanation": "Mrs. Cratchit is depicted as..."}	{ "title": "The Cratchit Family and Chris..."}	2026-01-19	5
3	6930bed7d7734e0b94af0a6bdd944d2e	26	26	2	20	[]	Bob Cratchit and His Family on Christmas Day	This community centers around Bob Cratchit, hi...	# Bob Cratchit and His Family on Christmas Day...	6.5	The impact severity rating is moderate, reflec...	{ "explanation": "Bob Cratchit is depicted as..."}	{ "title": "Bob Cratchit and His Family o..."}	2026-01-19	7
4	0e58707941394379b26dca4552ea8a7f	27	27	2	22	[]	The Cratchit Family and Tiny Tim's Fate	This community centers around the Cratchit fam...	# The Cratchit Family and Tiny Tim's Fate\n...	7.5	The impact severity rating is high due to the ...	{ "explanation": "Bob Cratchit is depicted as..."}	{ "title": "The Cratchit Family and Tiny ...}	2026-01-19	4

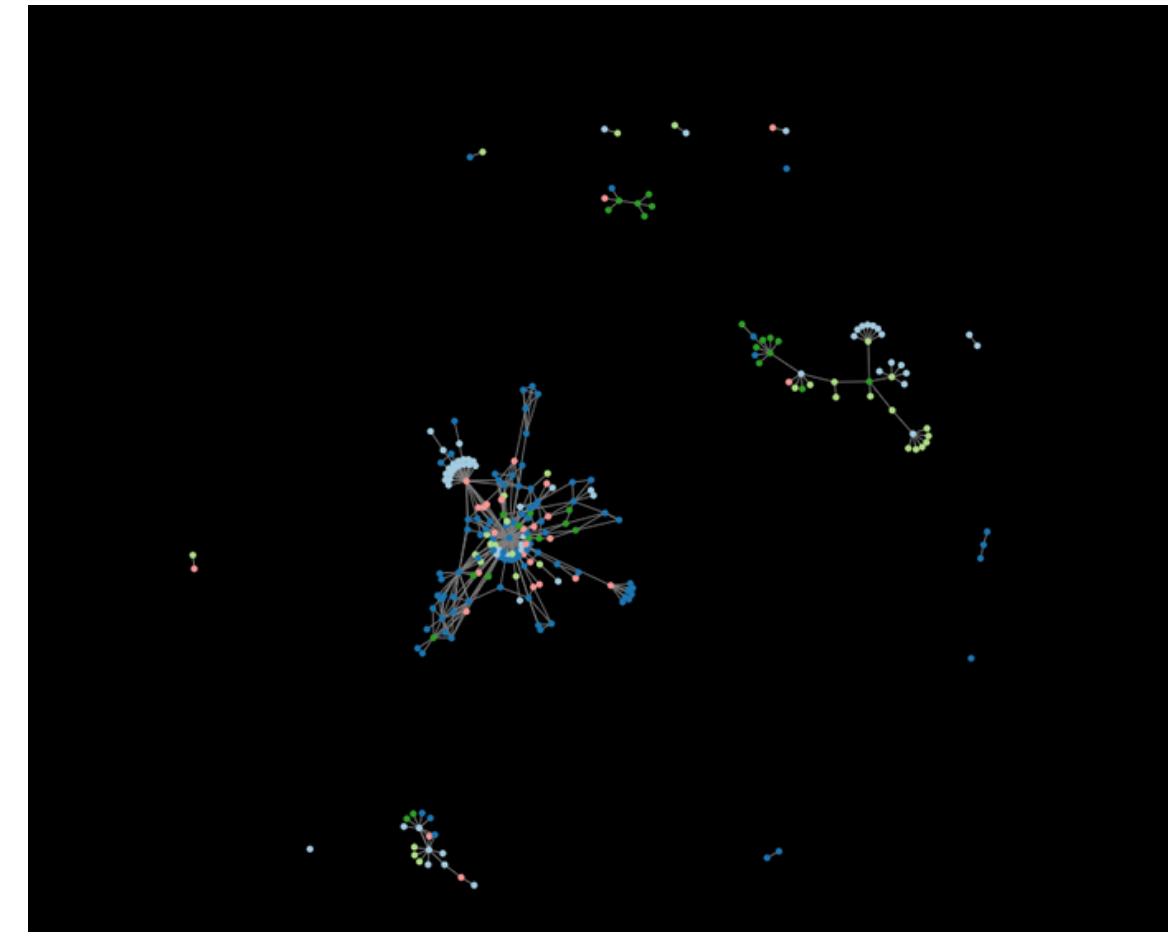
communities_reports.parquet

Graph RAG - experiment

Visualize indexing Graph on <https://noworneverev.github.io/graphrag-visualizer/>



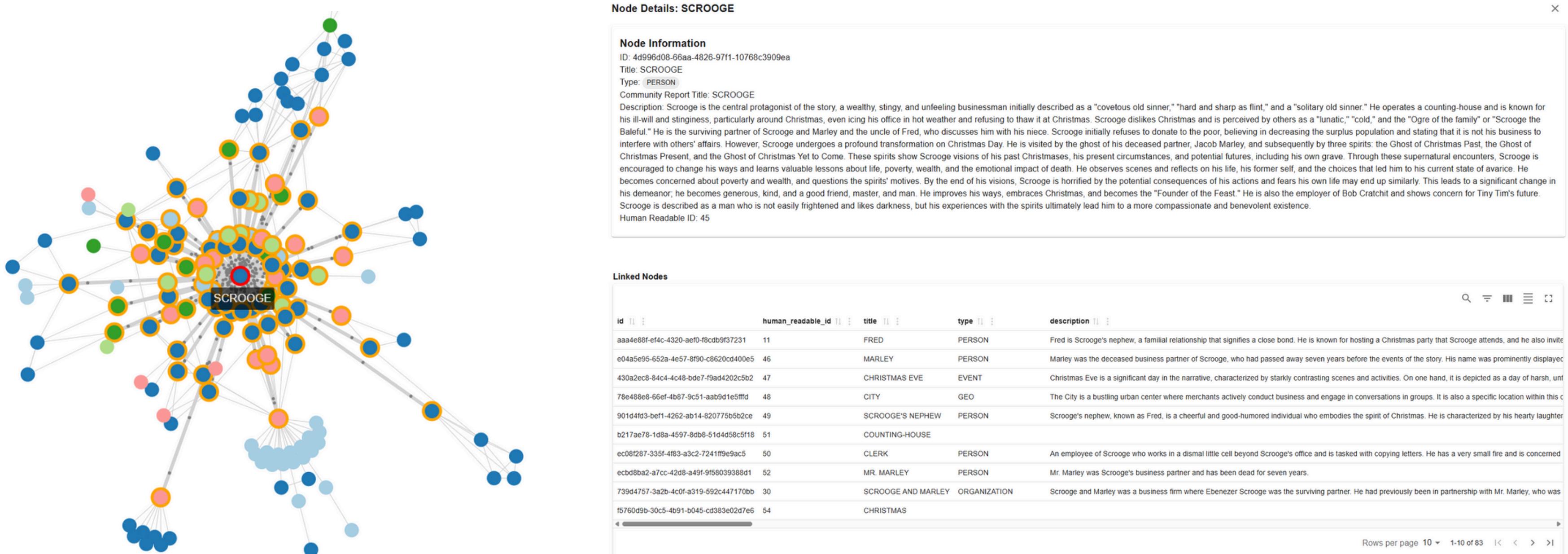
.parquet files



Graph visualization

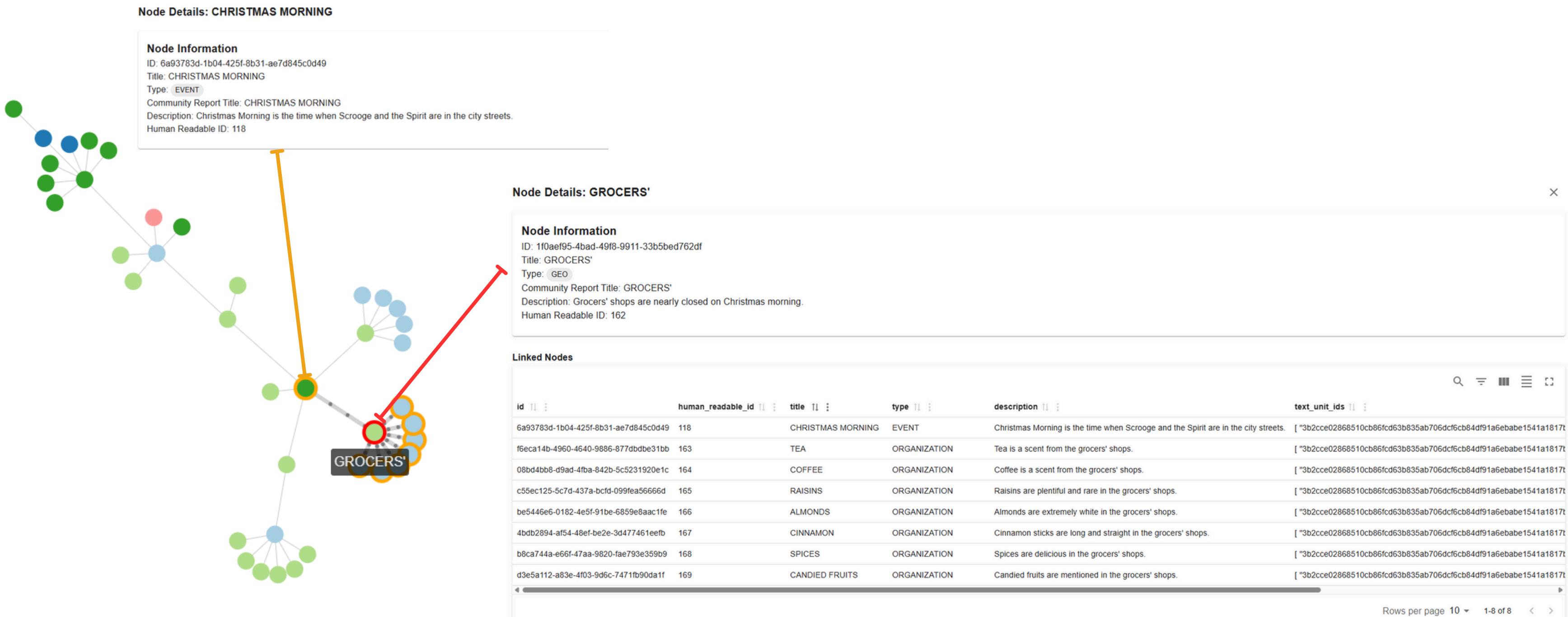
Graph RAG - experiment

Visualize indexing Graph on <https://noworneverev.github.io/graphrag-visualizer/>

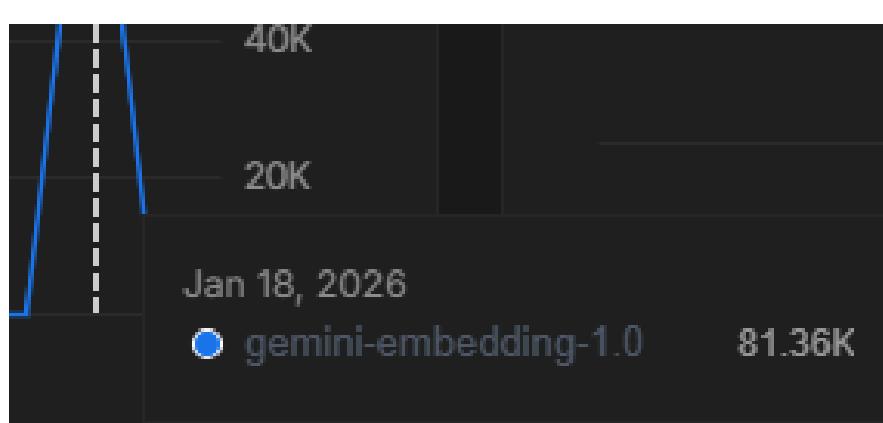
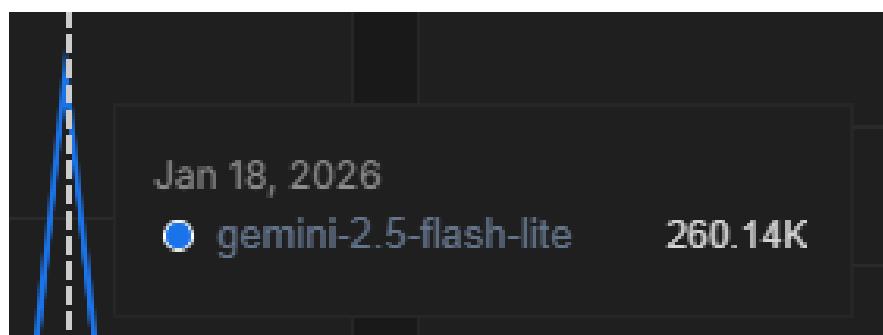
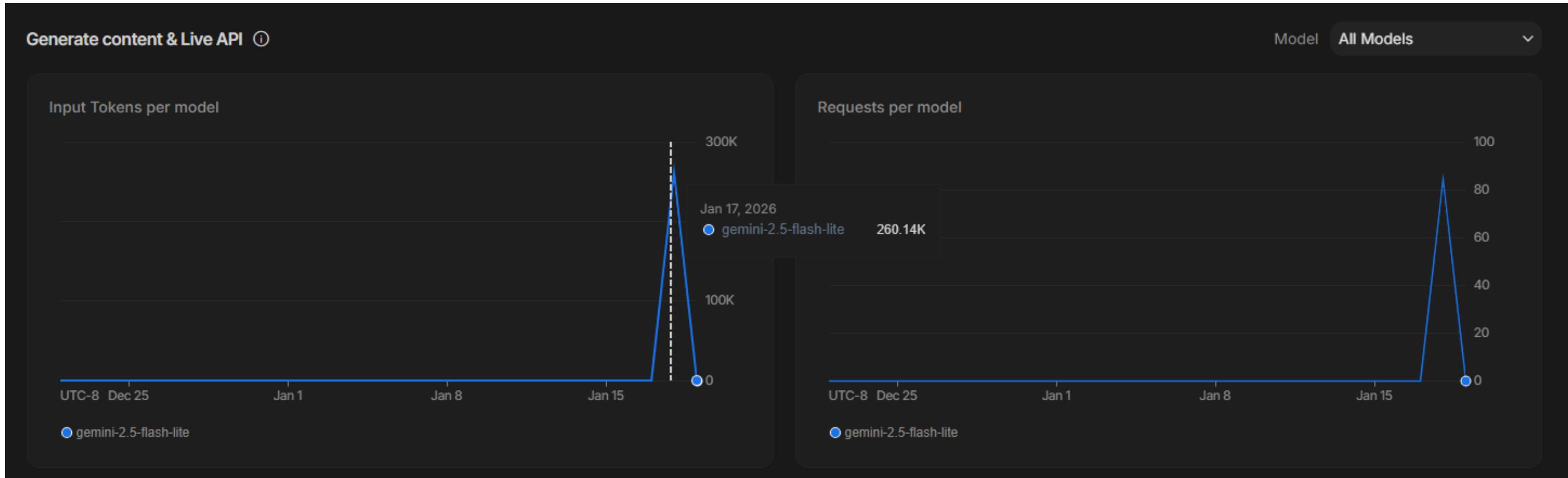


Graph RAG - experiment

Visualize indexing Graph on <https://noworneverev.github.io/graphrag-visualizer/>



Graph RAG - experiment



GraphRAG consumes a lot of LLM resources!

**Input Tokens per model (chat): 260.14k
Embedding: 81.36k**

Graph RAG - experiment

Global search

```
context_builder = GlobalCommunityContext(  
    community_reports=reports,  
    communities=communities,  
    entities=entities, # default to None if you don't want to use community weights for ranking  
    tokenizer=tokenizer,  
)
```

```
search_engine = GlobalSearch(  
    model=model,  
    context_builder=context_builder,  
    tokenizer=tokenizer,  
    max_data_tokens=5000, # change this based on the token limit you have on your model (if you are using a model with 8k limit, a good setting could be 5000)  
    map_llm_params=map_llm_params,  
    reduce_llm_params=reduce_llm_params,  
    allow_general_knowledge=False, # set this to True will add instruction to encourage the LLM to incorporate general knowledge in the response, which may incr  
    json_mode=True, # set this to False if your LLM model does not support JSON mode.  
    context_builder_params=context_builder_params,  
    concurrent_coroutines=32,  
    response_type="single paragraph", # free form text describing the response type and format, can be anything, e.g. prioritized list, single paragraph, multip  
)
```

Q: What are the top themes in this story?

Ans (1st paragraph)

LLM calls: 5. Prompt tokens: 29323. Output tokens: 2417.

The central theme of the narrative is the profound transformation of Ebenezer Scrooge, a miserly and unfeeling businessman, into a generous and kind individual. This significant change is primarily driven by supernatural encounters on Christmas Day with the ghosts of Jacob Marley and the Three Spirits of Christmas (Past, Present, and Yet to Come), who reveal visions of his past, present, and future, ultimately orchestrating his change of heart and understanding [Data: Reports (11, 9, 2, 1, 13, 14)]. Christmas itself serves as a pivotal theme, acting as the backdrop and catalyst for Scrooge's transformation and the story's events, encompassing traditions, food, decorations, and social gatherings, which starkly contrast with Scrooge's initial disdain for the holiday [Data: Reports (2, 11)]. The theme of redemption is central, as Scrooge's encounters with the spirits force him to confront his past actions and the consequences of his miserly life, leading to his eventual redemption and embrace of the Christmas spirit [Data: Reports (11, 2)].

Graph RAG - experiment

Local search

```
context_builder = LocalSearchMixedContext(  
    community_reports=reports,  
    text_units=text_units,  
    entities=entities,  
    relationships=relationships,  
    # if you did not run covariates during indexing, set this to None  
    covariates=None,  
    entity_text_embeddings=description_embedding_store,  
    embedding_vectorstore_key=EntityVectorStoreKey.ID, # if the vectorstore uses entity title as ids, set this to EntityVectorStoreKey.TITLE  
    text_embedder=text_embedder,  
    tokenizer=tokenizer,  
)
```

Relationships

Q: Who is Scrooge and what are his main relationships?

Ans (single paragraph)

LLM calls: 1. Prompt tokens: 7016. Output tokens: 988.

```
Ebenezer Scrooge, also known as Scrooge, is the central protagonist of the narrative. He is initially depicted as a wealthy, stingy, and unfeeling businessman, described as a "covetous old sinner" and "hard and sharp as  
## Key Relationships  
Scrooge's life is marked by several significant relationships, though many are strained by his disposition.  
### Family  
* **Fred (Scrooge's Nephew):** Fred is Scrooge's nephew and a close familial relation [Data: Entities (11)]. He is known for his cheerful and good-humored nature, embodying the spirit of Christmas [Data: Entities (49)]  
* **Fan (Scrooge's Sister):** Fan is Scrooge's younger sister. She is noted for coming to bring him home from school for Christmas [Data: Entities (26)].  
* **Scrooge's Niece:** Married to Fred, Scrooge's niece is described as a pretty and enthusiastic individual with a hearty laugh and strong opinions [Data: Entities (193)]. She shows interest in Scrooge's wealth and i  
### Business and Associates  
* **Bob Cratchit:** Bob Cratchit is Scrooge's loyal clerk, enduring a meager salary of fifteen shillings a week [Data: Entities (7)]. Despite his low wages, Bob Cratchit is a devoted family man. Scrooge is his employe  
* **Jacob Marley (Deceased Partner):** Jacob Marley was Scrooge's business partner, who had been deceased for seven years [Data: Entities (52, 46)]. His name was prominently displayed on their warehouse door [Data: En  
* **Fezziwig:** Fezziwig was Scrooge's former employer when Scrooge was young [Data: Relationships (99)].  
* **Dutch Merchant:** The Dutch merchant built the fireplace in Scrooge's house [Data: Relationships (67)].  
### Other Notable Figures  
* **Belle:** Belle is depicted as Scrooge's former fiancée and an old sweetheart. She appears to Scrooge in a vision of his past, described as a comely matron [Data: Entities (20)].  
* **Topper:** Topper is a bachelor and a friend of Scrooge's nephew and niece. He is implied to have a romantic interest in one of Scrooge's nieces' sisters and is skilled in singing and playing blind man's buff [Data  
Scrooge's journey is one of profound transformation, moving from a state of isolation and avarice to one of generosity and compassion, largely influenced by his interactions with supernatural entities and his reflection
```

Graph RAG - Use for ?

GraphRAG is able to **connect information across large volumes of information and use these connections to answer questions** that are difficult or impossible to answer using keyword and vector-based search mechanisms. Building on the previous question, provide semi-technical, high-level information on how the system offers functionality for various uses.

"**This lets a system using GraphRAG to answer questions where the answers span many documents as well as thematic questions such as “what are the top themes in this dataset?”**

Ref: <https://github.com/microsoft/graphrag>



References

NotebookLM Link: <https://notebooklm.google.com/notebook/7843f990-ffd0-4f91-9f87-228cf48010a>

RAG techniques IBM: <https://www.ibm.com/think/topics/rag-techniques>

2025's Ultimate Guide to RAG Retrieval: How to Pick the Right Method

: <https://medium.com/@mehulpratapsingh/2025s-ultimate-guide-to-rag-retrieval-how-to-pick-the-right-method-and-why-your-ai-s-success-2cedcda99f8a>

The 2025 Guide to Retrieval-Augmented Generation (RAG)

: <https://www.edenai.co/post/the-2025-guide-to-retrieval-augmented-generation-rag>

Top 13 Advanced RAG Techniques for Your Next Project

: <https://www.edenai.co/post/the-2025-guide-to-retrieval-augmented-generation-rag>