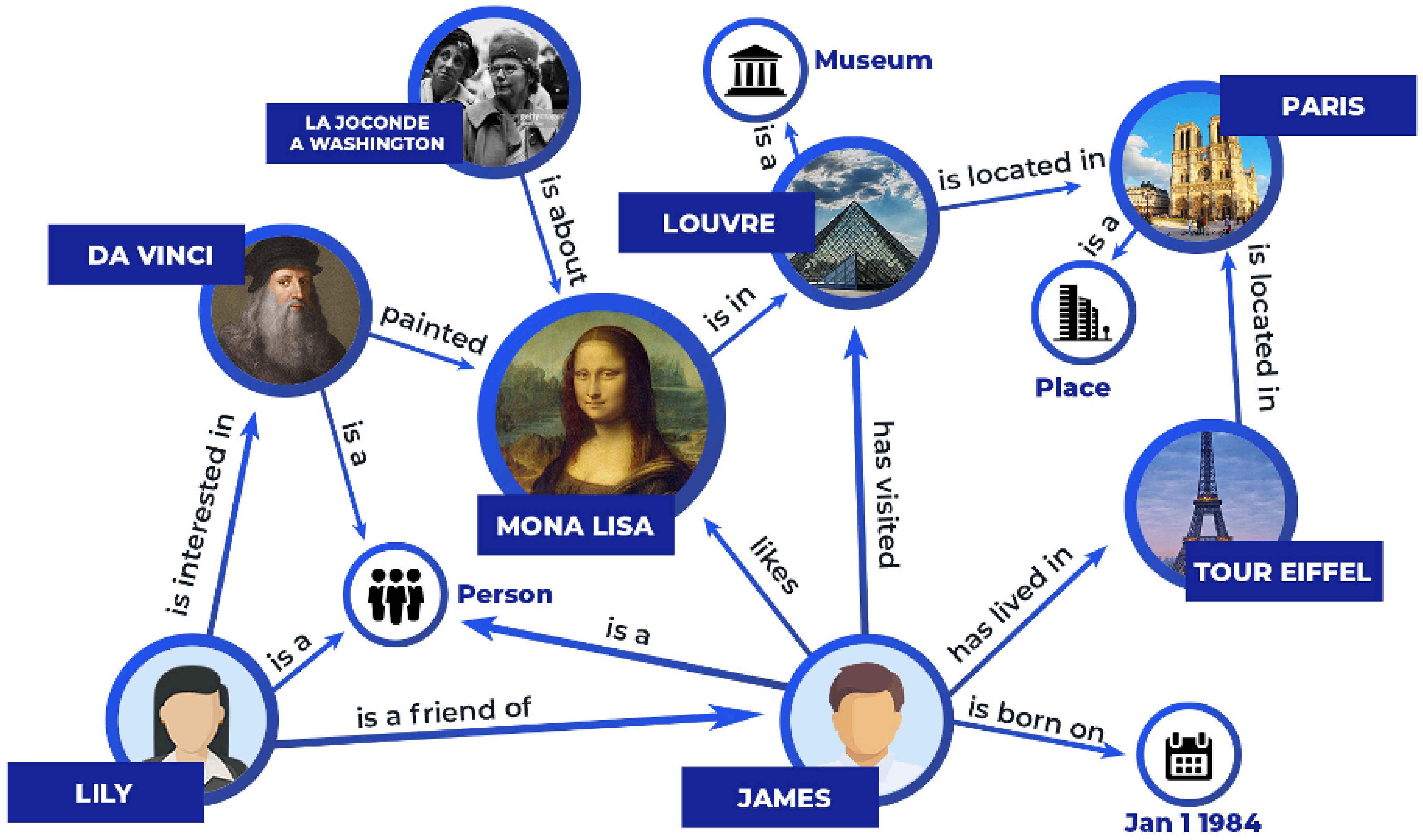
Incorporating KGs to Enhance LLM Performance

Máté Vass

Knowledge graphs



Drawbacks of LLMs

- Not explainable
- Lack of real understanding
- No domain knowledge
- Not up-to-date
- Hallucinations

Retrieval Augmented Generation

- Add knowledge to LLM context
  - Context window is a limit
- Model editing
  - Computanional cost

- Extracting relevant knowledge is crucial!

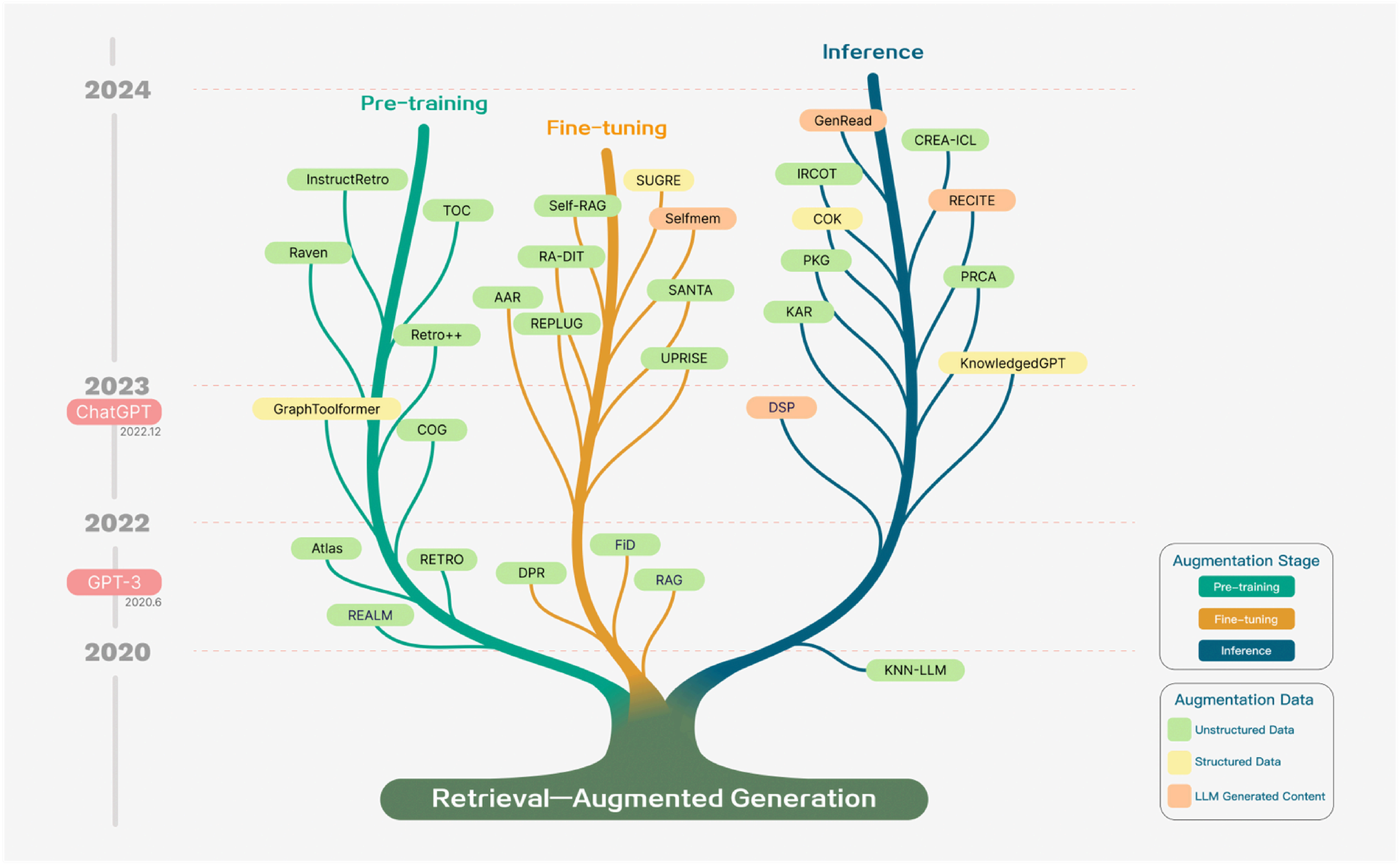


Figure 1: A timeline of existing RAG research. The timeline was established mainly according to the release date.

Naive RAG Pipeline

- Data indexing (text)
- Chunking (split)
- Embedding (chunk - vector pairs)
- Retrieve relevant chunks
- Generate answer using extracted data

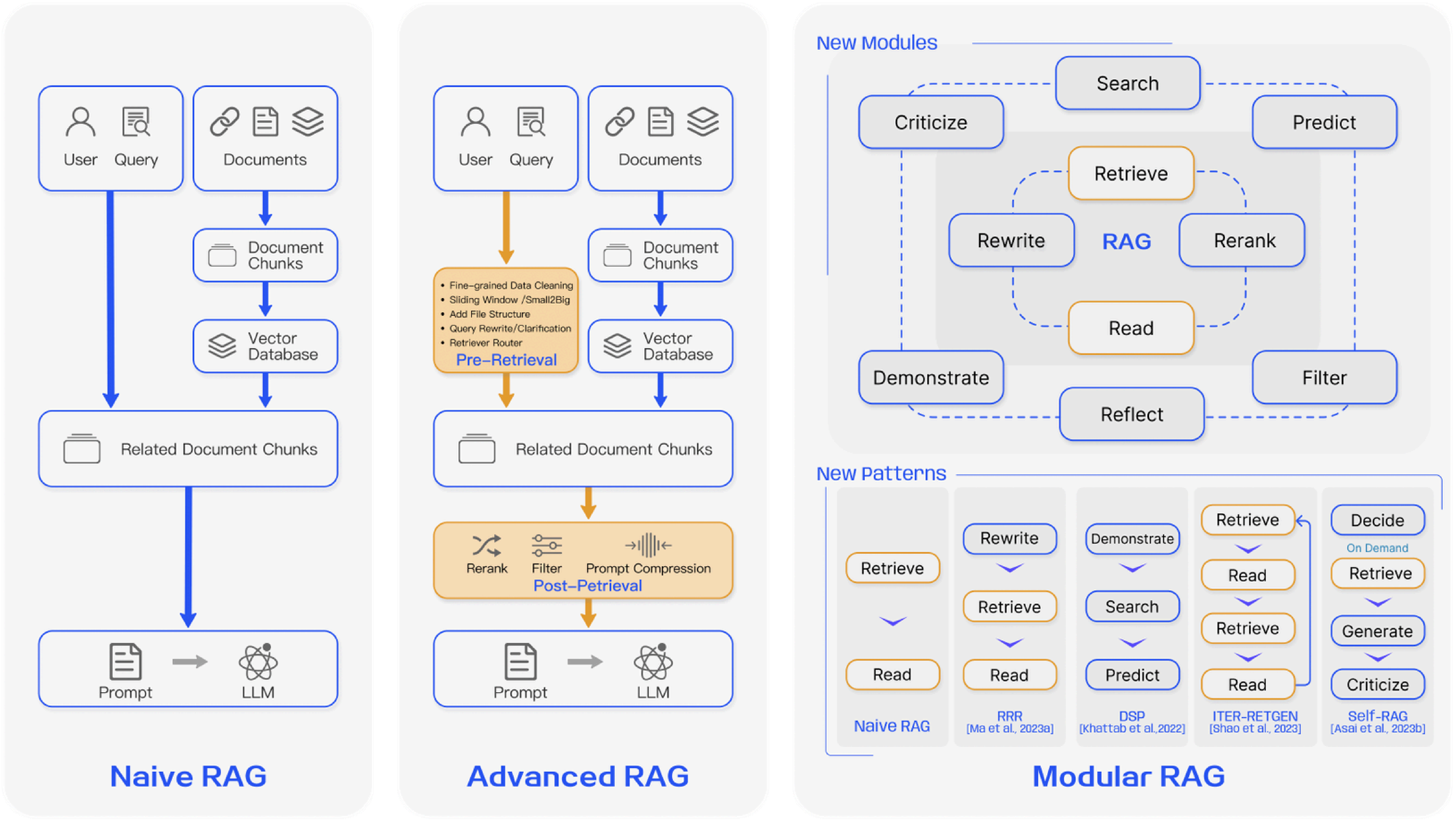
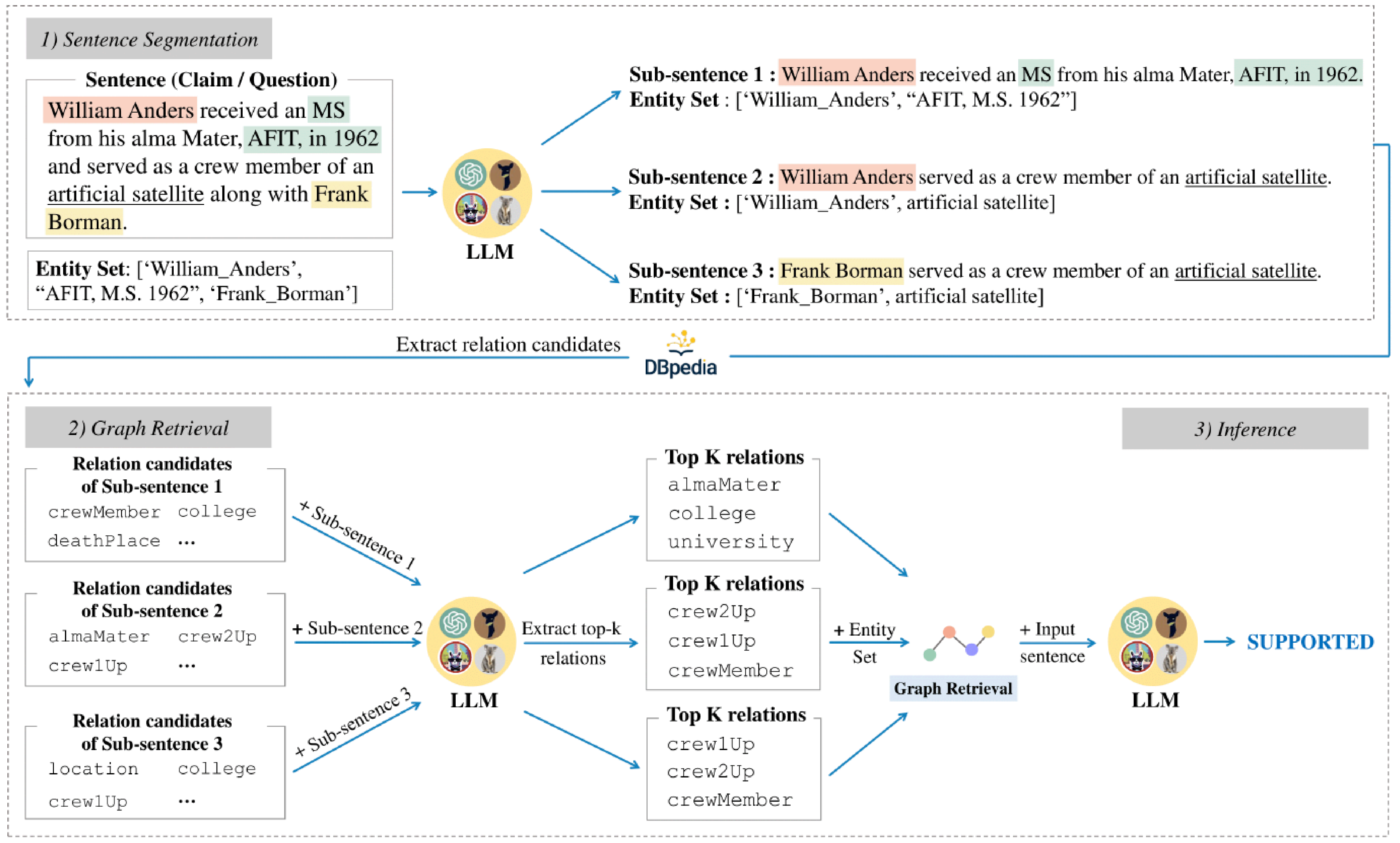


Figure 3: Comparison between the three paradigms of RAG

Examples

- KG is already available
    - KG-GPT
    - KELP
- Textual documents
    - GraphRAG

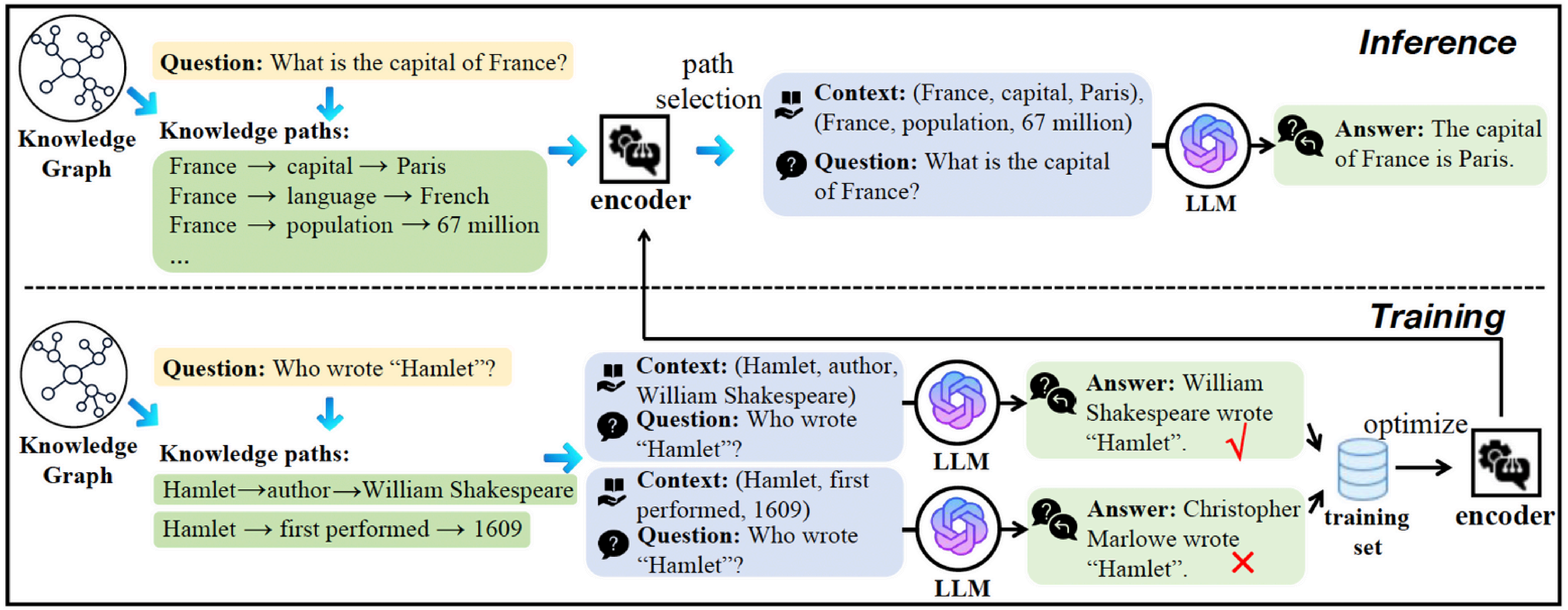
KG-GPT



1) Sentence Segmentation
Sentence (Claim / Question)
William Anders received an MS from his alma Mater, AFIT, in 1962 and served as a crew member of an artificial satellite along with Frank Borman.
Entity Set: ['William_Anders', "AFIT, M.S. 1962", 'Frank_Borman']
LLM
Sub-sentence 1 : William Anders received an MS from his alma Mater, AFIT, in 1962.
Entity Set : ['William_Anders', "AFIT, M.S. 1962"]
Sub-sentence 2 : William Anders served as a crew member of an artificial satellite.
Entity Set : ['William_Anders', artificial satellite]
Sub-sentence 3 : Frank Borman served as a crew member of an artificial satellite.
Entity Set : ['Frank_Borman', artificial satellite]
Extract relation candidates
DBpedia
2) Graph Retrieval
Relation candidates of Sub-sentence 1
crewMember college deathPlace ...
Relation candidates of Sub-sentence 2
almaMater crew2Up crew1Up ...
Relation candidates of Sub-sentence 3
location college crew1Up ...
+ Sub-sentence 1
+ Sub-sentence 2
+ Sub-sentence 3
LLM
Extract top-k relations
Top K relations
almaMater college university
Top K relations
crew2Up crew1Up crewMember
Top K relations
crew1Up crew2Up crewMember
+ Entity Set
Graph Retrieval
+ Input sentence
3) Inference
LLM
SUPPORTED

Figure 1: An overview of KG-GPT. The framework comprises three distinct phases: Sentence Segmentation, Graph Retrieval, and Inference. The given example comes from FACTKG. It involves a 2-hop inference from 'William_Anders' to 'Frank_Borman', requiring verification through an evidence graph consisting of three triples. Both 'William_Anders' and 'Frank_Borman' serve as internal nodes in DBpedia (Lehmann et al., 2015), while "AFIT, M.S. 1962" acts as a leaf node. Moreover, *artificial satellite* represents the *Type* information absent from the provided entity set.

KELP



Inference
Knowledge Graph
Question: What is the capital of France?
Knowledge paths:
France ⟶ capital ⟶ Paris
France ⟶ language ⟶ French
France ⟶ population ⟶ 67 million
...
path selection
encoder
Context: (France, capital, Paris), (France, population, 67 million)
Question: What is the capital of France?
LLM
Answer: The capital of France is Paris.
Training
Knowledge Graph
Question: Who wrote "Hamlet"?
Knowledge paths:
Hamlet⟶author⟶William Shakespeare
Hamlet ⟶ first performed ⟶ 1609
Context: (Hamlet, author, William Shakespeare)
Question: Who wrote "Hamlet"?
Context: (Hamlet, first performed, 1609)
Question: Who wrote "Hamlet"?
LLM
LLM
Answer: William Shakespeare wrote "Hamlet". √
Answer: Christopher Marlowe wrote "Hamlet". ✗
optimize
training set
encoder

Figure 2: The overall pipeline of the proposed KELP. During the inference phase, we identify knowledge paths from the knowledge graph that are associated with the entities present in the input question. An encoder is then trained to select valuable paths as knowledge contexts. Finally, the selected knowledge contexts, along with the input question, are input into the LLM to generate the final answer.

GraphRAG

- Extracting a knowledge graph out of raw text
- Building a community hierarchy
- Generating summaries for these communities
- Using these structures for RAG-based tasks

GraphRAG Indexing

- Split text into chunks
- Extract entities, relations
- Hierarchical Leiden community detection
- Bottom-up summary generation
- Query: utilizes the resulting KG

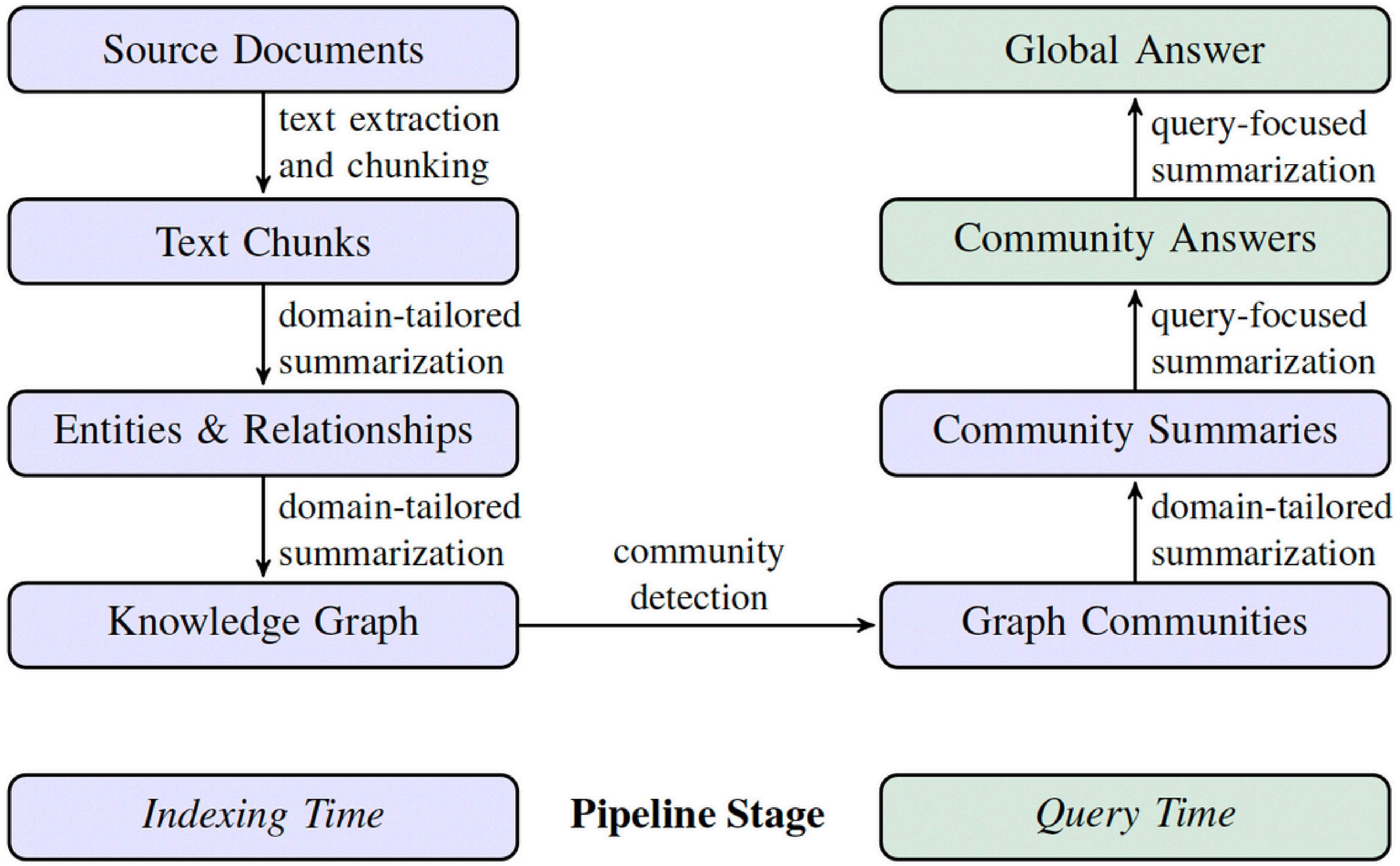


Figure 1: Graph RAG pipeline using an LLM-derived graph index of source document text. This graph index spans nodes (e.g., entities), edges (e.g., relationships), and covariates (e.g., claims) that have been detected, extracted, and summarized by LLM prompts tailored to the domain of the dataset. Community detection (e.g., Leiden, Traag et al., 2019) is used to partition the graph index into groups of elements (nodes, edges, covariates) that the LLM can summarize in parallel at both indexing time and query time. The “global answer” to a given query is produced using a final round of query-focused summarization over all community summaries reporting relevance to that query.

Thank you for your attention!

Questions?