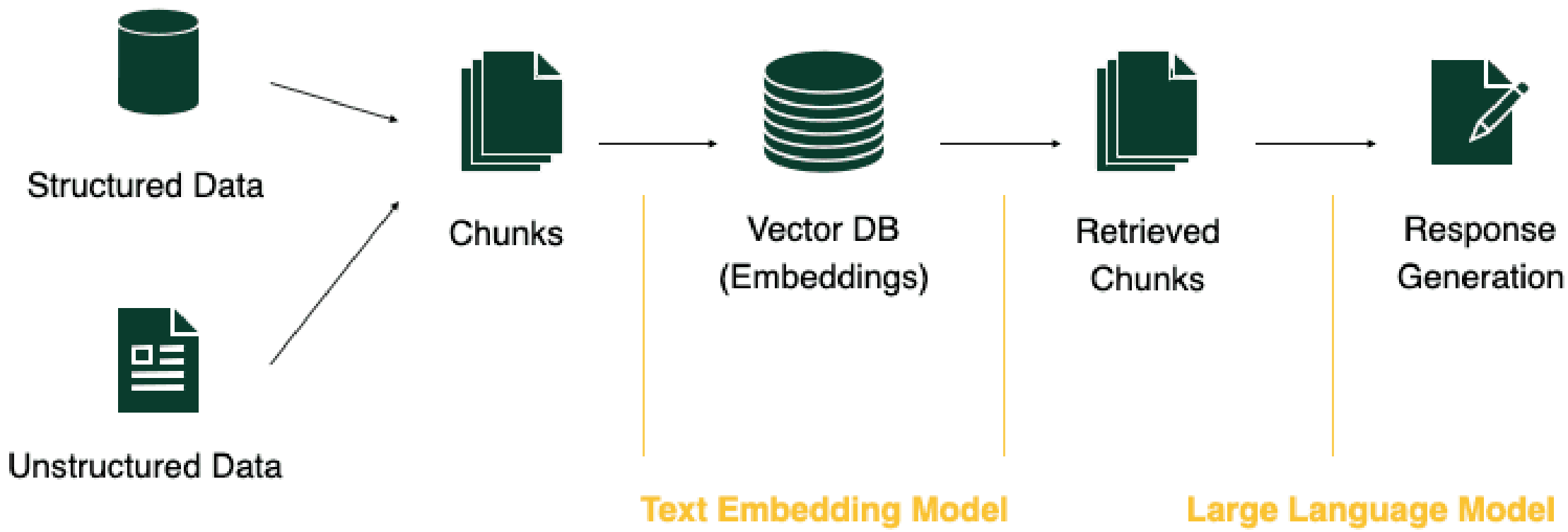
RAG Vs CAG

Everything that you need to know in simple terms

Bhavishya Pandit

Challenges of RAG



Source

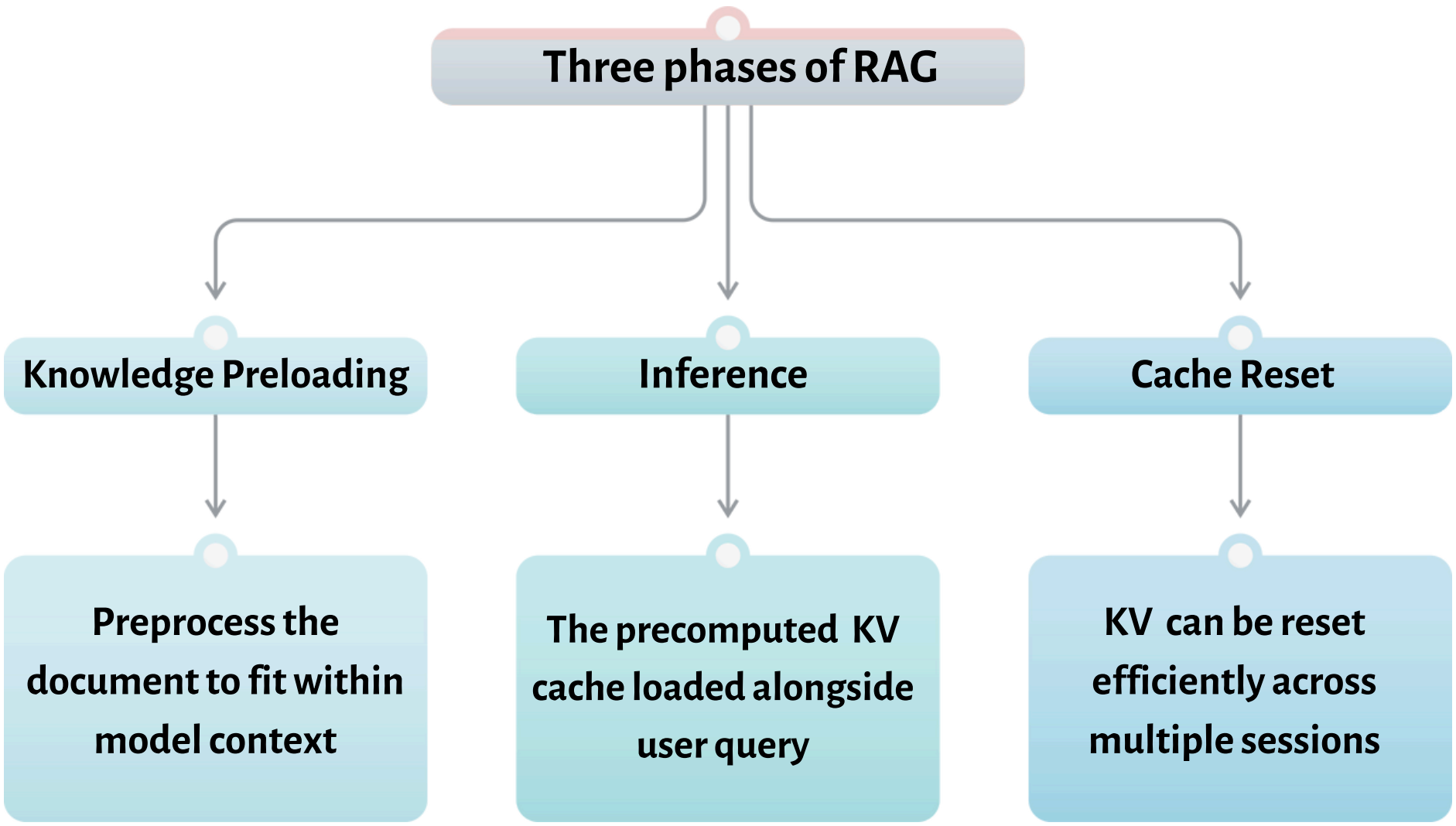
- **Retrieval Latency**: RAG relies on real-time retrieval of documents, which introduces delays, especially when handling large or complex knowledge bases.
- **Errors in Document Ranking**: Inaccurate or irrelevant document retrieval can lead to suboptimal answers, reducing the system's reliability.
- **Complicated Setup**: Integrating retrieval and generation components requires careful tuning, additional infrastructure, and ongoing maintenance, complicating workflows and increasing overhead.

These limitations highlight the need for alternatives like Cache-Augmented Generation (CAG) to streamline and optimize knowledge tasks.

Introducing CAG

Simpler, Faster, Better

Cache-Augmented Generation (CAG) eliminates the need for real-time document retrieval. Instead, it relies on preloaded knowledge and a Key-Value (KV) Cache to store essential information upfront.

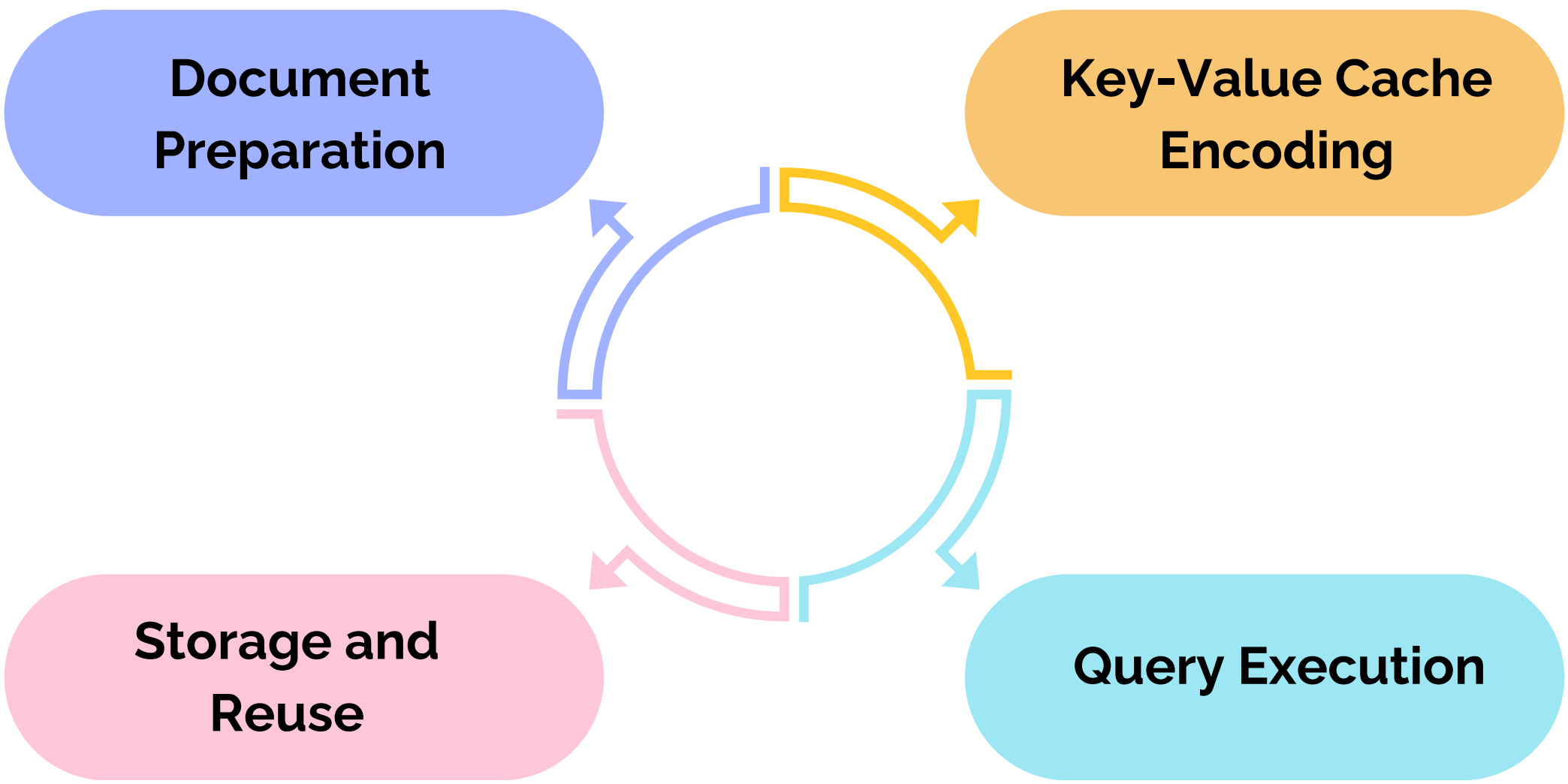


- **Preloaded knowledge**: All required documents are prepared and loaded into the model before inference.
- **Key-Value (KV) cache**: Documents are encoded into a Key-Value format during preprocessing. This cache stores inference states, to quickly access relevant data.
- **Context window**: The system uses the LLM's long context window to embed all relevant information directly, enabling seamless and accurate responses.
- **Query execution**: User queries directly interact with the preloaded data in the cache, ensuring instant, relevant answers without the delays of retrieval processes.
- **Storage and reuse**: The precomputed cache is stored in memory, making it reusable for multiple queries and significantly improving system efficiency.

Why Is CAG Retrieval-Free?

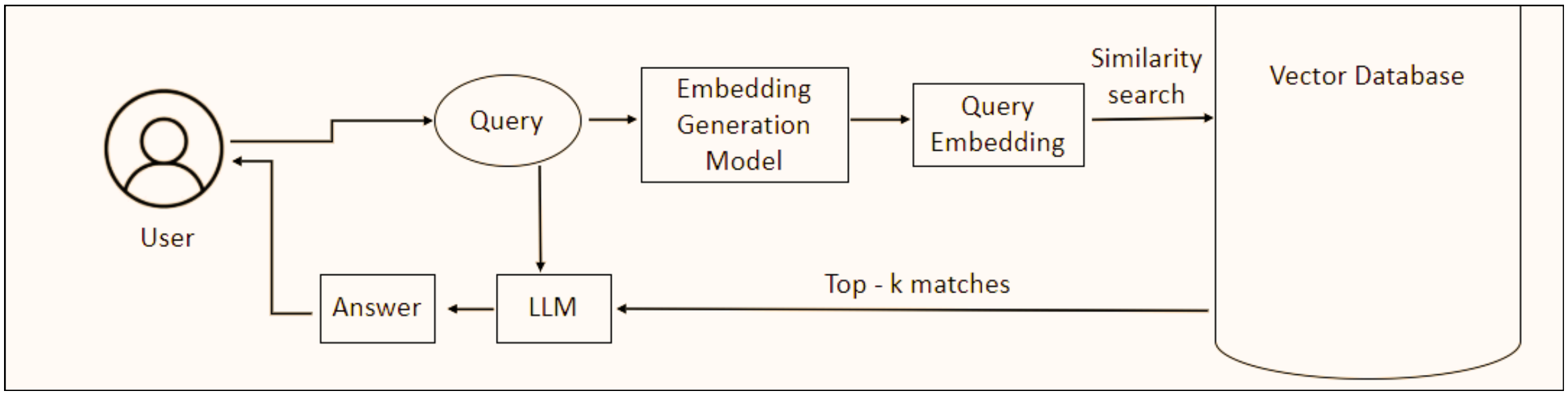
- Preloaded Knowledge
- Precomputed Memory
- Direct Access to Context
- Error-Free Responses

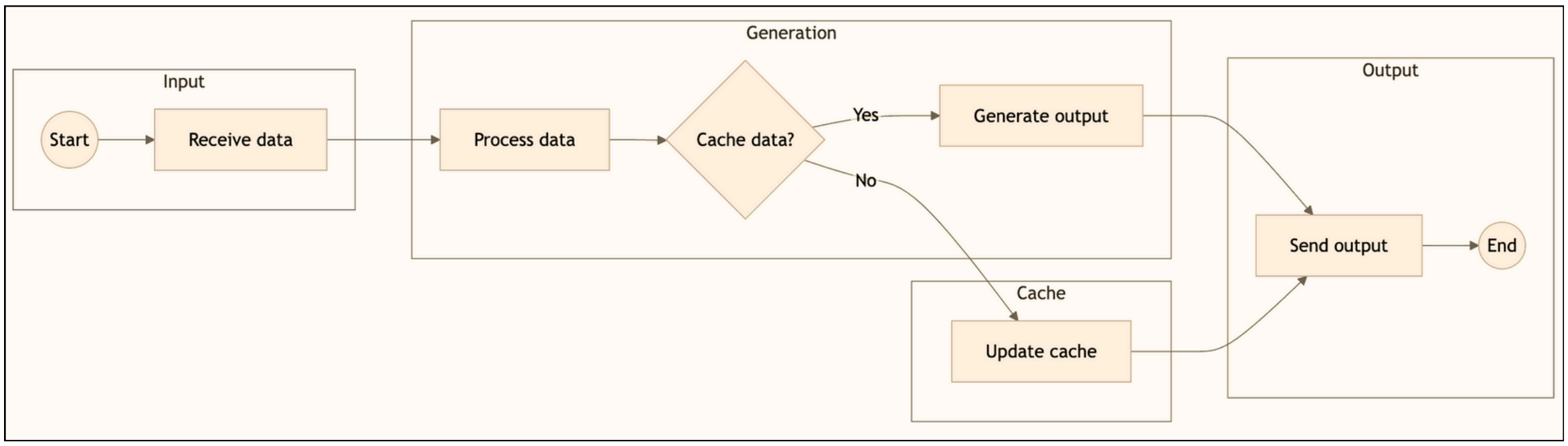
How Does CAG Preload Context?



Comparison of RAG & CAG Workflows

RAG





CAG

RAG workflow: The model relies on real-time retrieval during inference. When a query is made, it fetches relevant documents from external sources, which are then processed and incorporated into the response. This dynamic retrieval introduces latency and can lead to errors if the wrong documents are selected.

CAG workflow: CAG system preloads all required knowledge into a Key-Value (KV) cache before the query is made. During inference, the model can access this preloaded context directly, bypassing the need for any retrieval step. As a result, CAG is faster, simpler, and more accurate, eliminating the dependency on external data.

Experimental Analysis

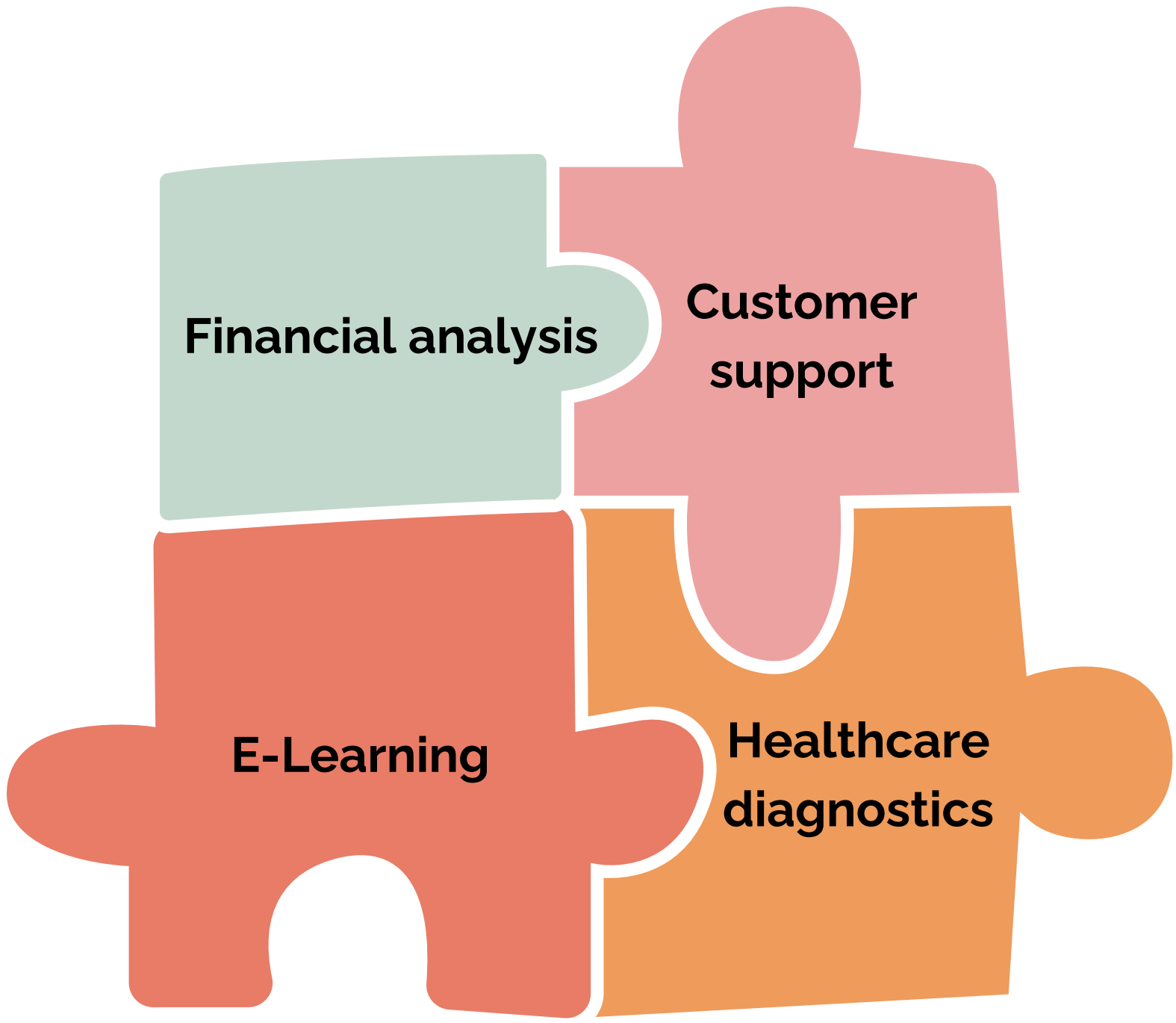
Table 3: Comparison of Generation Time

Dataset	Size	System	Generation Time (s)
HotpotQA	Small	CAG	0.85292
		w/o CAG	9.24734
	Medium	CAG	1.66132
		w/o CAG	28.81642
	Large	CAG	2.32667
		w/o CAG	94.34917
SQuAD	Small	CAG	1.06509
		w/o CAG	10.29533
	Medium	CAG	1.73114
		w/o CAG	13.35784
	Large	CAG	2.40577
		w/o CAG	31.08368

Source

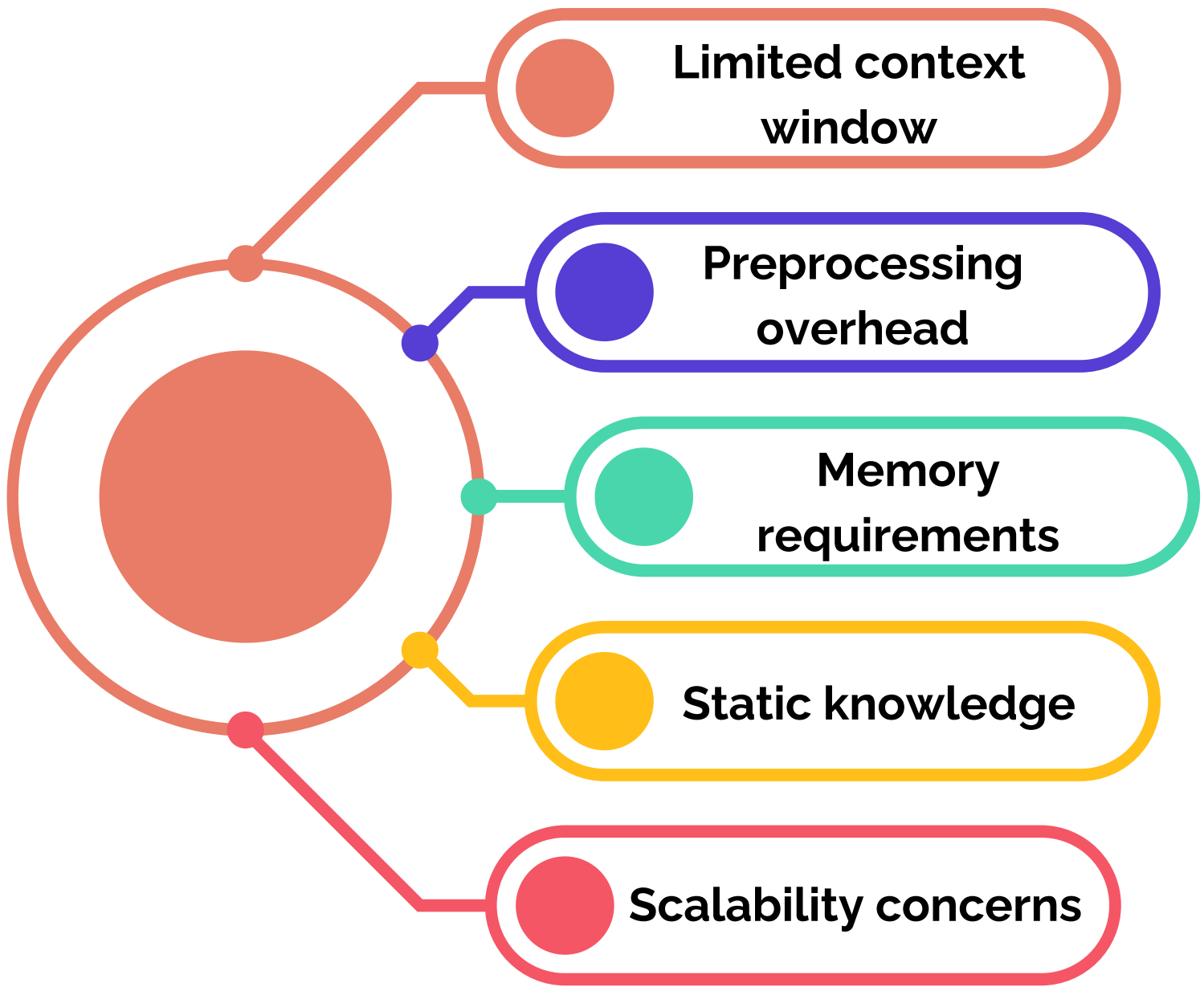
- **Accuracy and Response Quality:** CAG consistently outperformed sparse (BM25) and dense (OpenAI Indexes) RAG systems in accuracy, measured by BERTScore.
- **Generation Time**: CAG significantly reduces response generation time by eliminating the retrieval step. Results showed that CAG was up to **94% faster**.
- **Efficiency and Scalability**: CAG processes all knowledge in a single preloaded context, avoiding the iterative retrieval steps of RAG.
- **System Complexity**: CAG features a simplified architecture by removing the need for retrieval and ranking components.

Real Life Usecases



- **Financial analysis**: By preloading market trends and financial data, CAG enables real-time, data-driven insights, streamlining decision-making processes.
- **Customer support**: CAG improves customer service by preloading FAQs and product documentation, ensuring quick, relevant and accurate responses to customer queries.
- **Healthcare diagnostics**: CAG can preload medical knowledge bases, enabling instant and accurate diagnostic support for healthcare professionals.
- **E-Learning**: CAG enhances learning experiences by preloading course materials and academic resources, allowing for instant assistance for students and educators.

Challenges of CAG



- **Limited context window**: CAG relies on preloading required data within the model's context window, this limitation can constrain the amount of usable information.
- **Preprocessing overhead**: Preparing, encoding & storing knowledge in a Key-Value cache requires significant effort during the setup, which may be time consuming.
- **Memory & Storage requirements**: Maintaining a large KV cache for extensive datasets can demand substantial memory or storage, increasing infrastructure costs.
- **Static knowledge**: Once preloaded, the system cannot dynamically adapt to updated information, making it unsuitable for scenarios requiring rapidly changing data.
- **Scalability concerns**: Scaling CAG for multiple simultaneous queries may introduce complexity in managing cache efficiency and resource allocation.

Follow to stay updated on Generative AI

SAVE

LIKE

REPOST