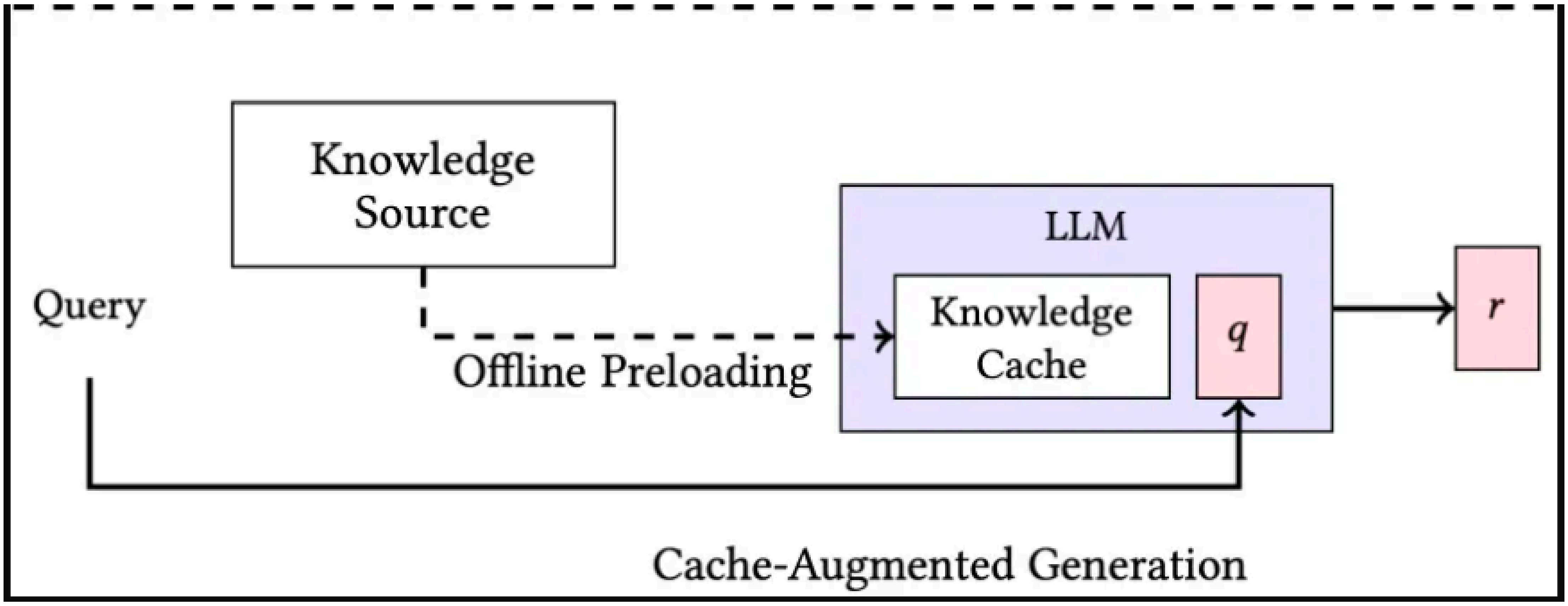
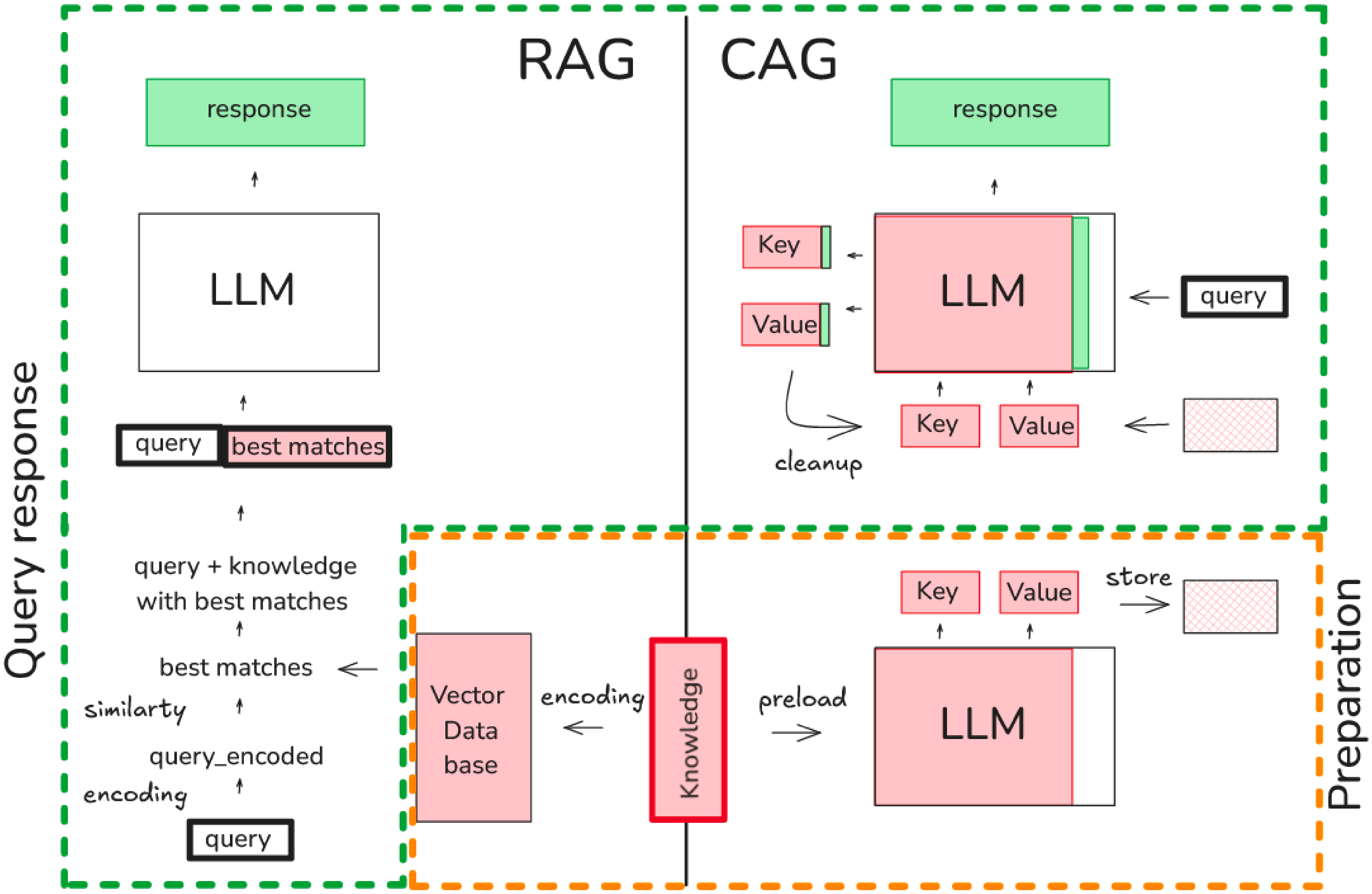
Analytics Vidhya

A Perfect Guide to
Cache-Augmented Generation (CAG)





What is CAG?

Cache-Augmented Generation (CAG) is an approach that enhances language models by preloading relevant knowledge into their context window, eliminating the need for real-time retrieval. CAG optimizes knowledge-intensive tasks by leveraging precomputed key-value (KV) caches, enabling faster and more efficient responses.

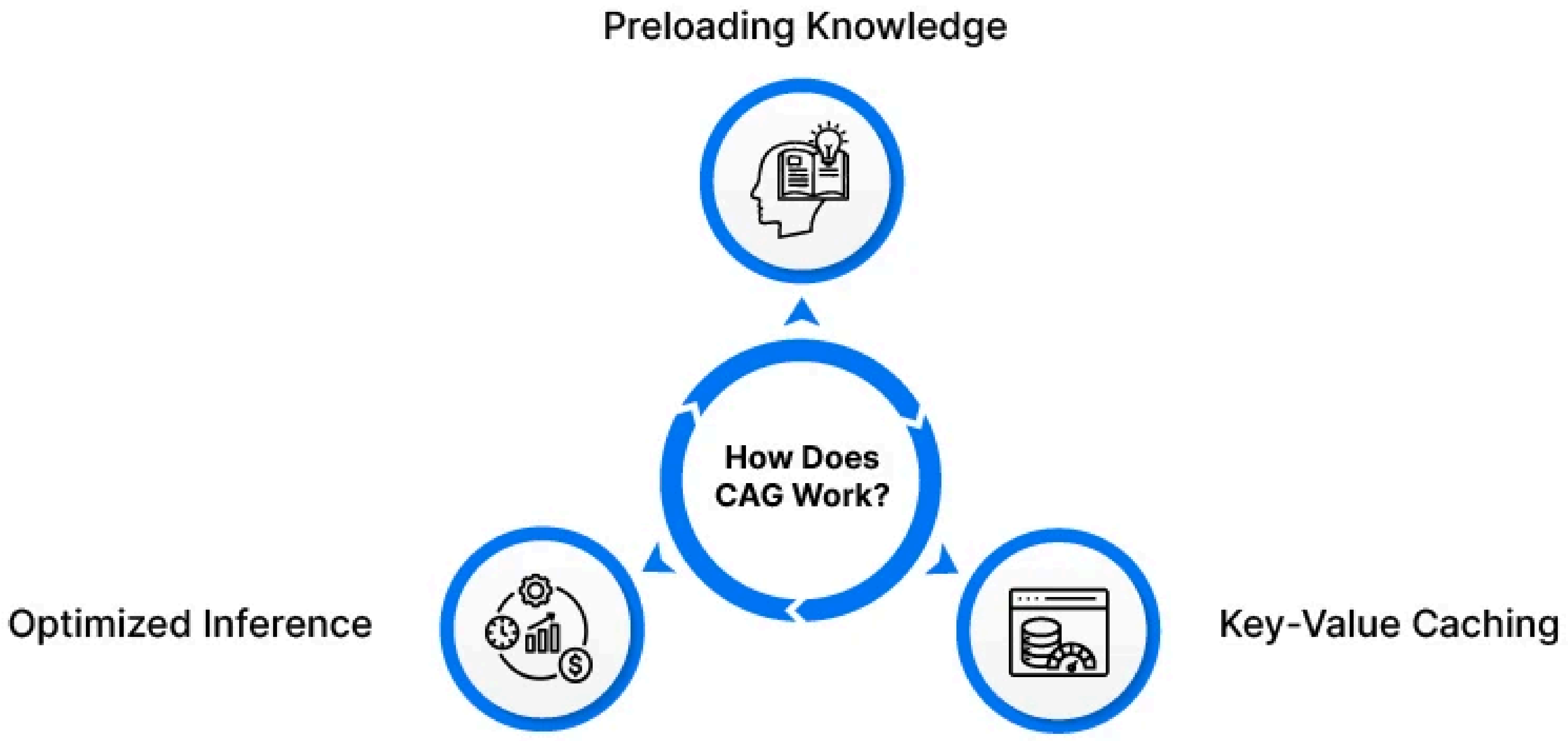
How Does CAG Work?

When a query is submitted, CAG follows a structured approach to retrieve and generate responses efficiently:

- **Preloading Knowledge**: Before inference, the relevant information is preprocessed and stored within an extended context or a dedicated cache.

- **Key-Value Caching**: Instead of dynamically fetching documents like RAG, CAG utilizes precomputed inference states. These states act as a reference, allowing the model to access cached knowledge instantly, bypassing the need for external lookups.

- **Optimized Inference**: When a query is received, the model checks the cache for pre-existing knowledge embeddings. If a match is found, the model directly utilizes the stored context to generate a response. This dramatically reduces inference time while ensuring coherence and fluency in generated outputs.



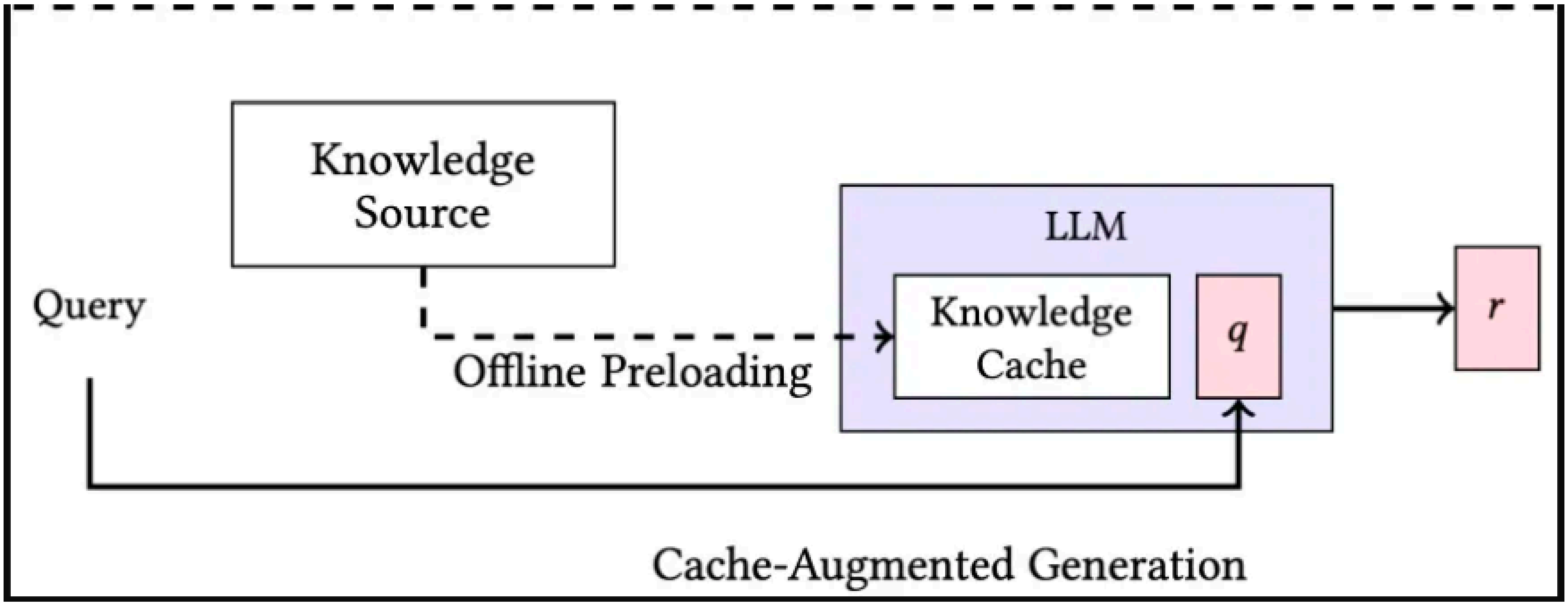
Key Differences from RAG

This is how CAG approach is different from RAG:

- **No real-time retrieval**: The knowledge is preloaded instead of being fetched dynamically.
- **Lower latency**: Since the model does not query external sources during inference, responses are faster.
- **Potential Staleness**: Cached knowledge may become outdated if not refreshed periodically.

CAG Architecture

To efficiently generate responses without real-time retrieval, CAG relies on a structured framework designed for fast and reliable information access. CAG systems consist of the following components:

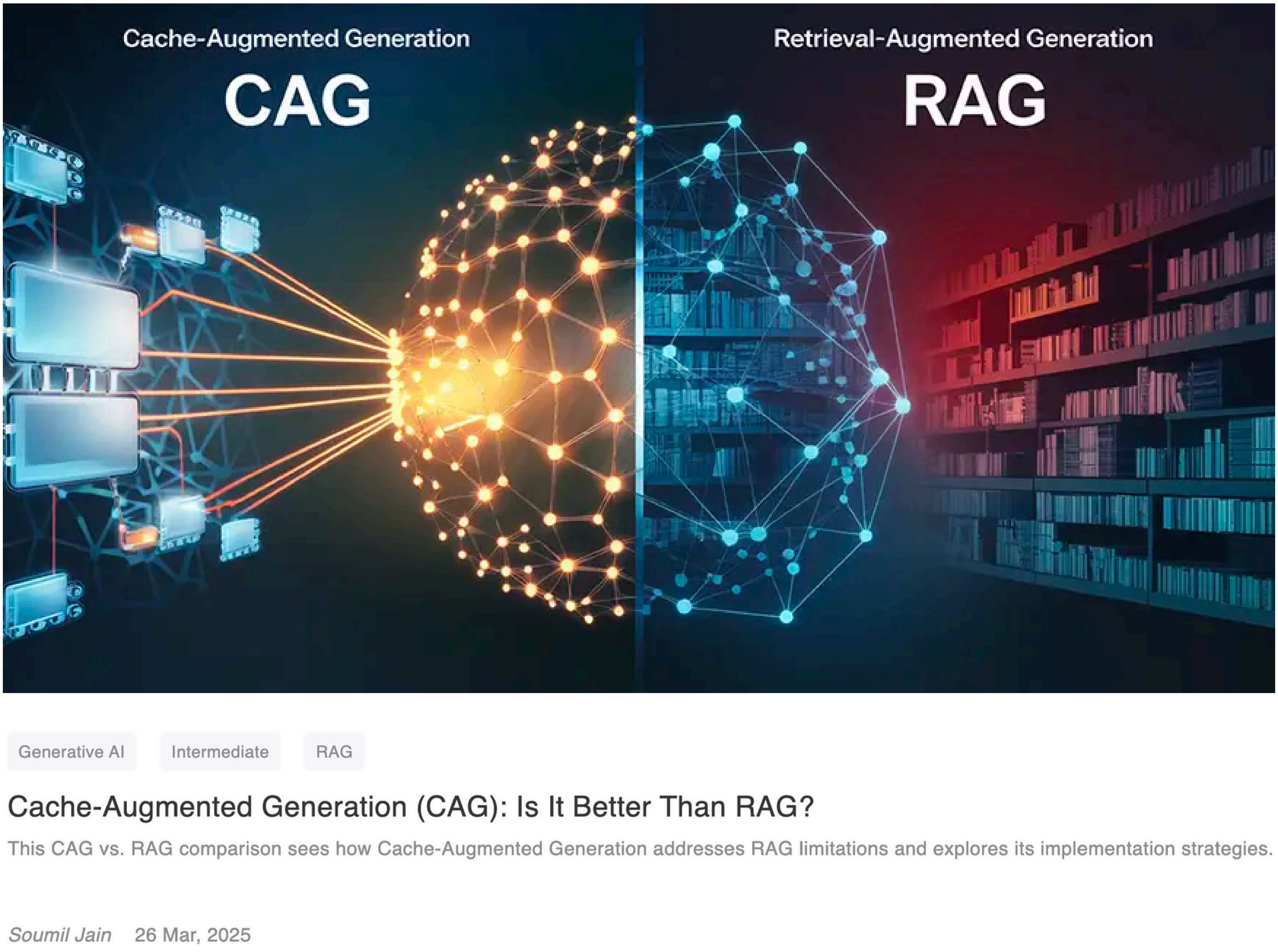


- **Knowledge Source**: A repository of information, such as documents or structured data, accessed before inference to preload knowledge.
- **Offline Preloading**: Knowledge is extracted and stored in a Knowledge Cache inside the LLM before inference, ensuring fast access without live retrieval.
- **LLM (Large Language Model)**: The core model that generates responses using preloaded knowledge stored in the Knowledge Cache.
- **Query Processing**: When a query is received, the model retrieves relevant information from the Knowledge Cache instead of making real-time external requests.

- **Response Generation**: The LLM produces an output using the cached knowledge and query context, enabling faster and more efficient responses.

This architecture is best suited for use cases where knowledge does not change frequently and fast response times are required.

For more information, visit this article



Generative AI Intermediate RAG

Cache-Augmented Generation (CAG): Is It Better Than RAG?

This CAG vs. RAG comparison sees how Cache-Augmented Generation addresses RAG limitations and explores its implementation strategies.

Soumil Jain 26 Mar, 2025