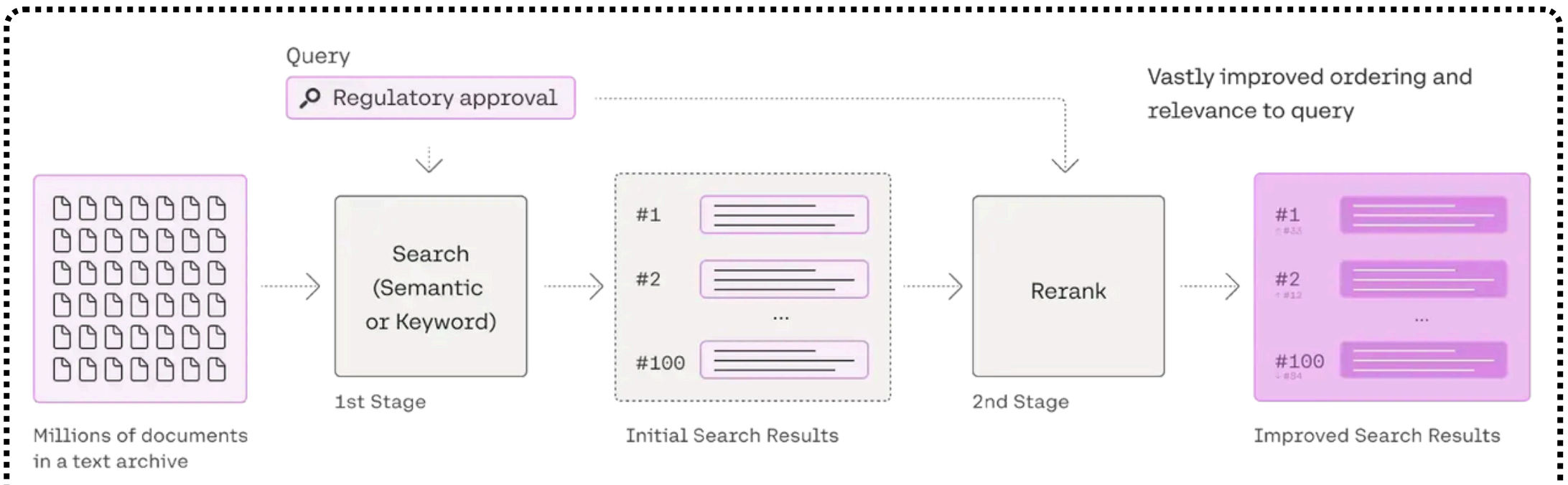
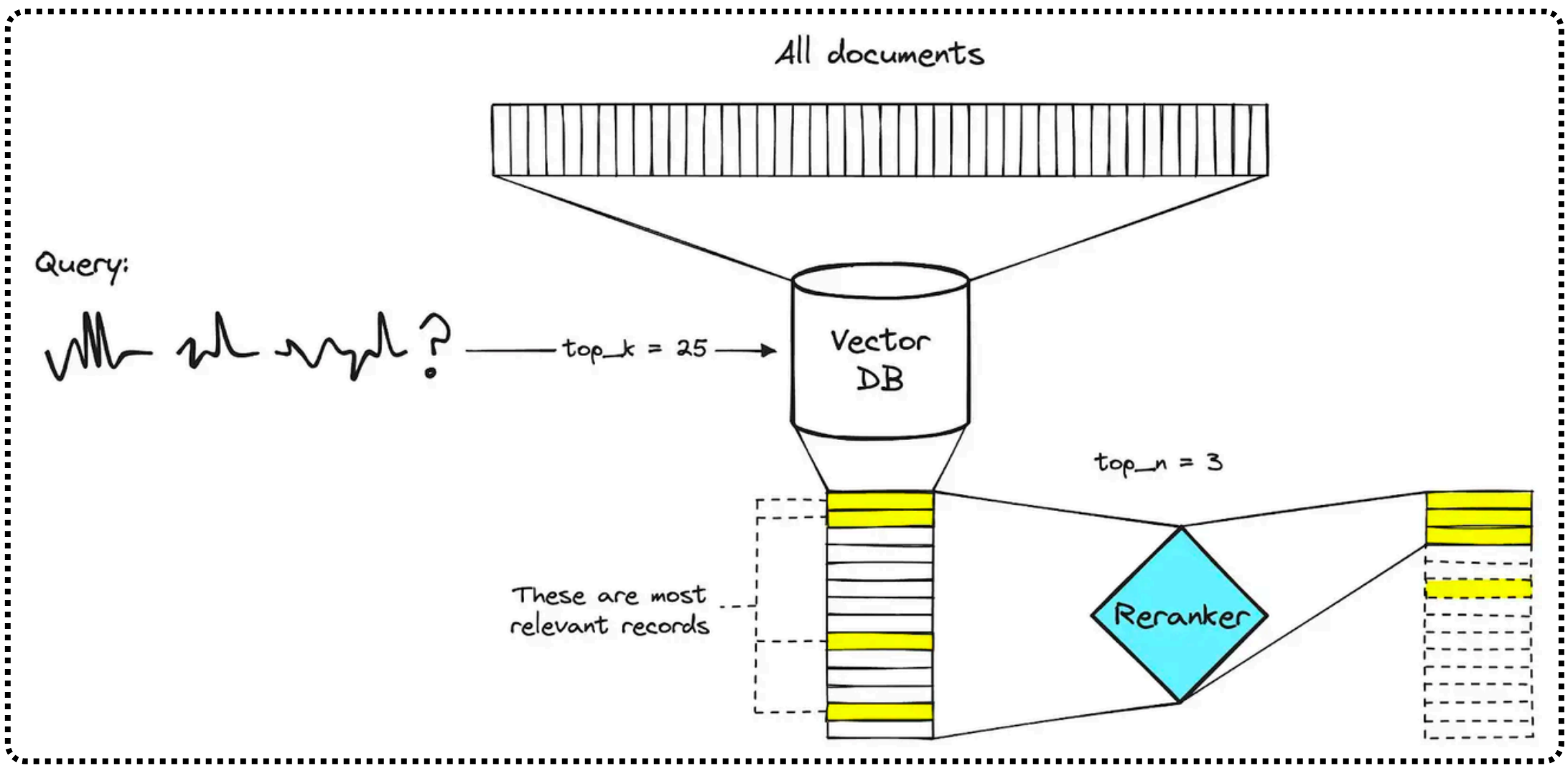
A Guide on Reranker for RAG



The Rerank endpoint acts as the last stage reranker of a search flow.



A two-stage retrieval system. The vector DB step will typically include a bi-encoder or sparse embedding model.

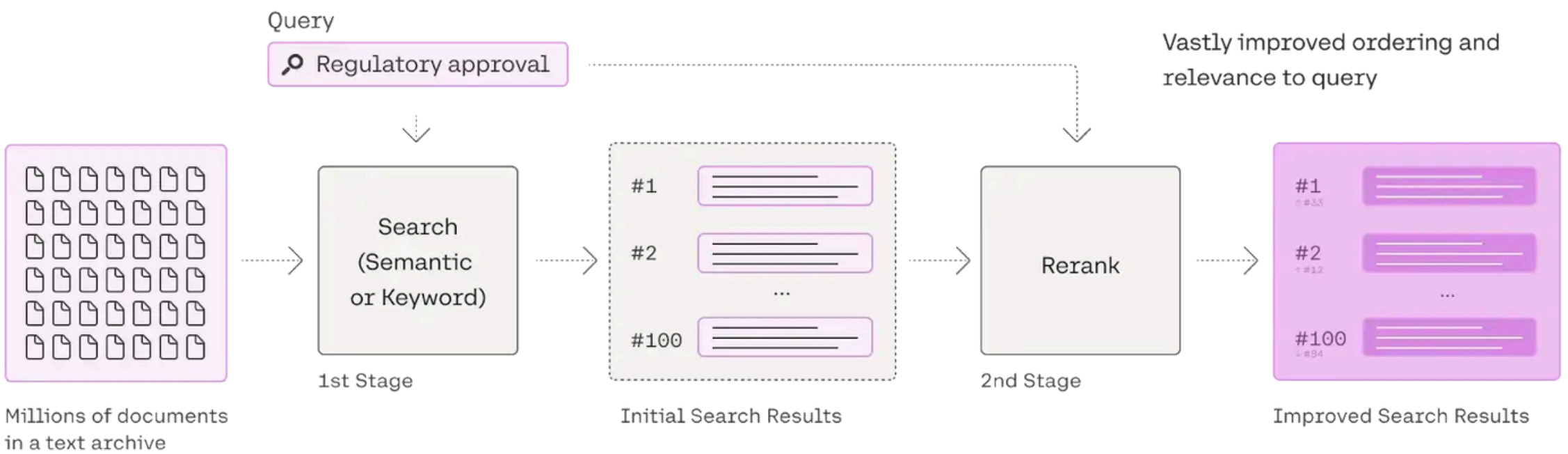
Retrieval Augmented Generation (RAG) systems are revolutionizing how we interact with information, but they're only as good as the data they retrieve. Optimizing those retrieval results is where the reranker comes in. For instance, consider it as a quality control system for your search results, ensuring that only the most relevant information comes into the final output.

This post explores the world of rerankers, explaining why they're important, when you need them, their potential drawbacks, and their types. This article will also guide you in selecting the best reranker for your specific RAG system and how to evaluate its performance.

What is a Reranker for RAG?

A reranker is an important component of the information retrieval systems, it acts as a second-pass filter. While doing an initial search (using methods like semantic or keyword search) it returns a set of documents, and the reranker helps to reorder them.

This reordering filters and prioritizes documents based on their relevance to a specific query, hence improving the quality of search results. Rerankers achieve this balance between speed and quality by employing more complex matching techniques than the initial retrieval stage.



This image illustrates a two-stage search process. Reranking is the second stage, where an initial set of search results, based on semantic or keyword matching, is refined to significantly improve the relevance and ordering of the final results, delivering a more accurate and useful outcome for the user's query.

Why to Use Reranker for RAG?

Imagine your RAG system as a chef, and the retrieved documents are the ingredients. To create a delicious dish (accurate answer), you require the best ingredients. But what if some of those ingredients are irrelevant or simply don't belong in the recipe?

That's where rerankers help!

Here's why you need a reranker:

- **Hallucination Reduction**: Rerankers filter out irrelevant documents that can cause the LLM to generate inaccurate or nonsensical responses (hallucinations).

- **Cost Savings**: By focusing on the most relevant documents, you reduce the amount of information the LLM needs to process, saving you money on API calls and computing resources.

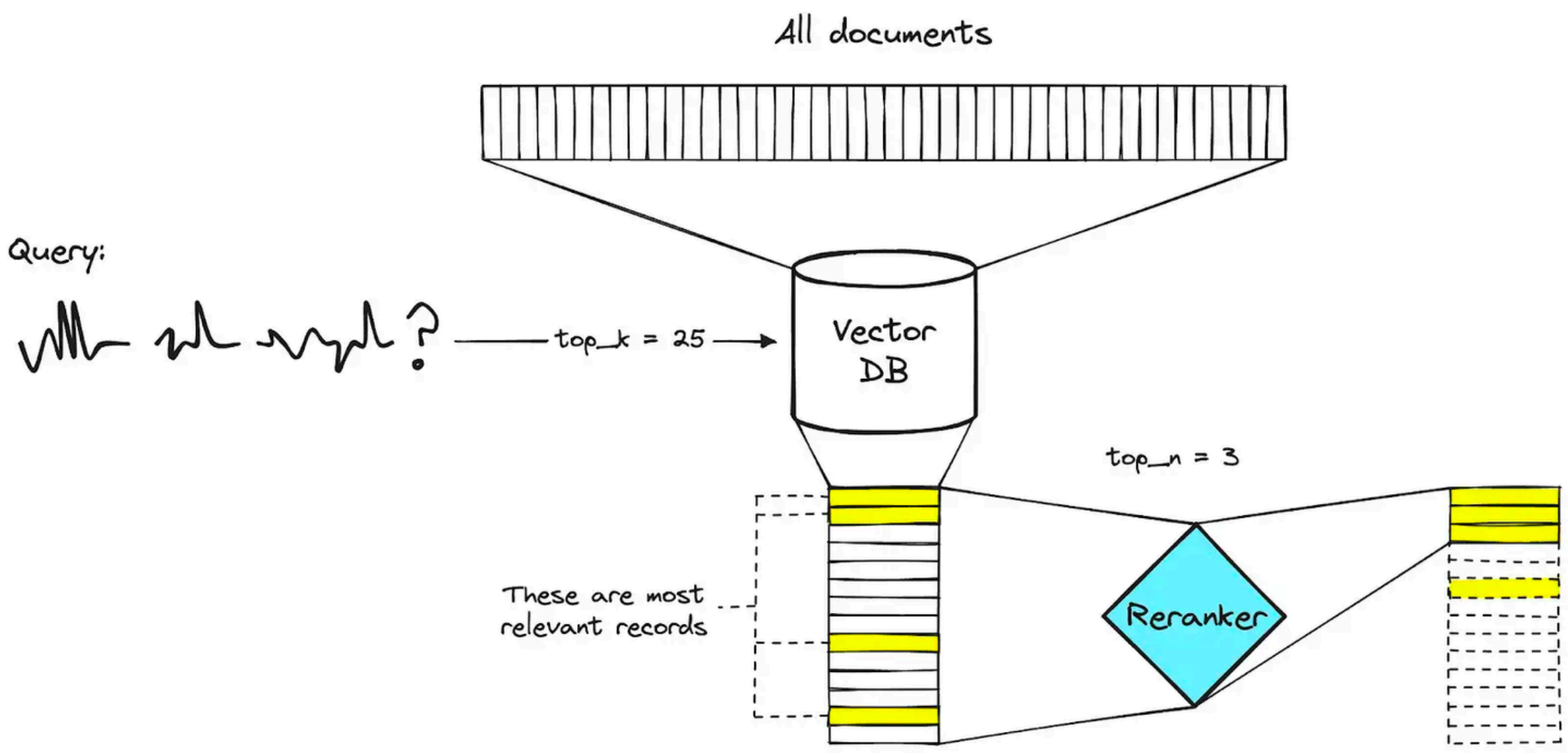
Rerankers Advantages

Rerankers excel where embeddings fall short by:

- **Bag-of-Embeddings Approach**: Breaking down documents into smaller, contextualized units of information instead of relying on a single vector representation.

- **Semantic Keyword Matching**: Combining the strengths of powerful encoder models (like BERT) with keyword-based techniques to capture both semantic meaning and keyword relevance.

- **Improved Generalization**: By focusing on smaller, contextualized units, rerankers handle unseen documents and queries more effectively.



A query is used to search a vector database, retrieving the top 25 most relevant documents. These documents are then passed to a “Reranker” module. The reranker refines the results, selecting the top 3 most relevant documents for the final output.

For more information, you can read this article