Semantic Search

Semantic Search means searching by meaning, not by exact words.

Instead of matching keywords, an LLM + embeddings system understands the context, intent, and relationships of words to find the most meaningful results.

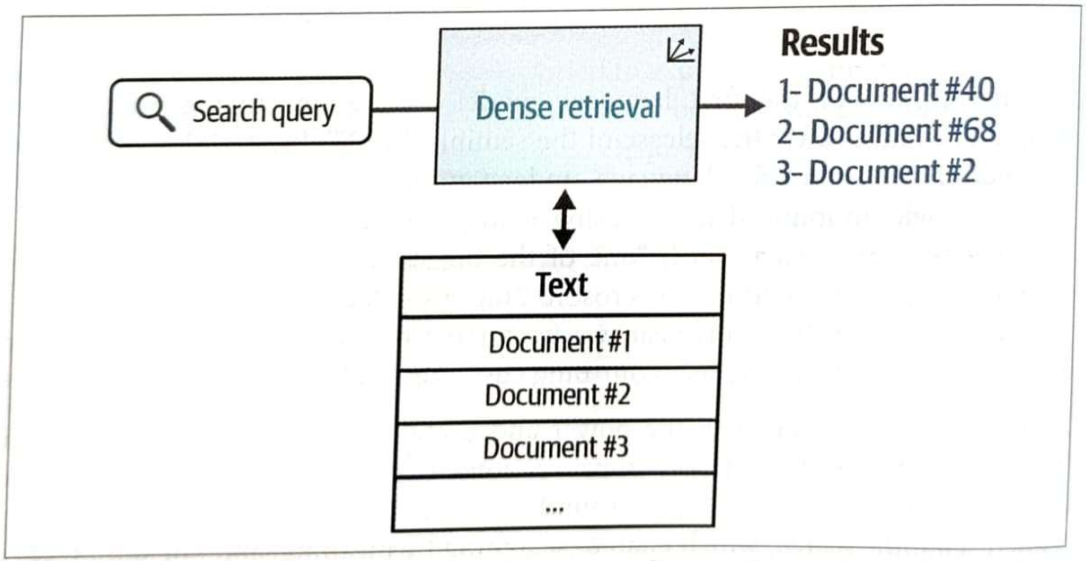

Figure 8-1. Dense retrieval is one of the key types of semantic search, relying on the similarity of text embeddings to retrieve relevant results.

Reranking

Search systems are often pipelines of multiple steps. A reranking language model is one of these steps and is tasked with scoring the relevance of a subset of results against the query; the order of results is then changed based on these scores. Figure 8-2 shows how rerankers are different from dense retrieval in that they take an additional input: a set of search results from a previous step in the search pipeline.

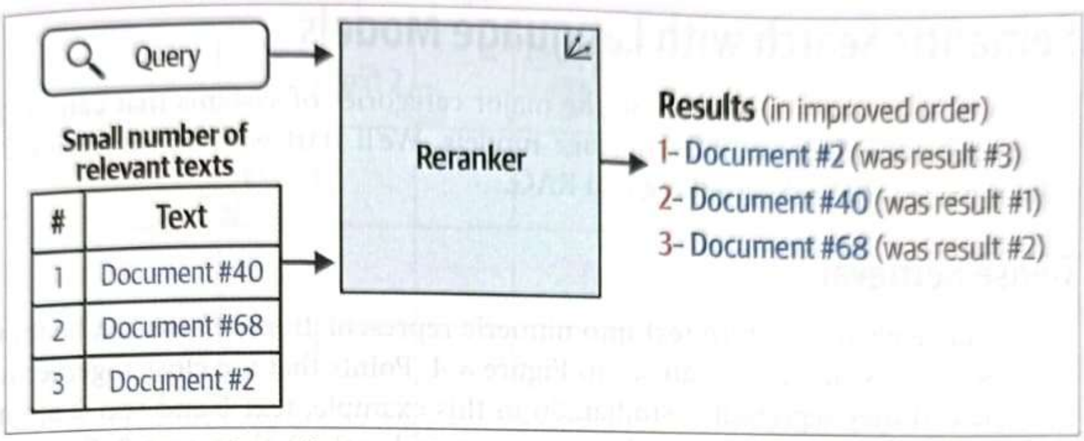
Query
Small number of relevant texts
#
Text
1
Document #40
2
Document #68
3
Document #2
Reranker
Results (in improved order)
1- Document #2 (was result #3)
2- Document #40 (was result #1)
3- Document #68 (was result #2)

Figure 8-2. Rerankers, the second key type of semantic search, take a search query and a collection of results, and reorder them by relevance, often resulting in vastly improved results.

RAG

The growing LLM capability of text generation led to a new type of search systems that include a model that generates an answer in response to a query. Figure 8-3 shows an example of such a generative search system.

Generative search is a subset of a broader type of category of systems better called RAG systems. These are text generation systems that incorporate search capabilities to reduce hallucinations, increase factuality, and/or ground the generation model on a specific dataset.

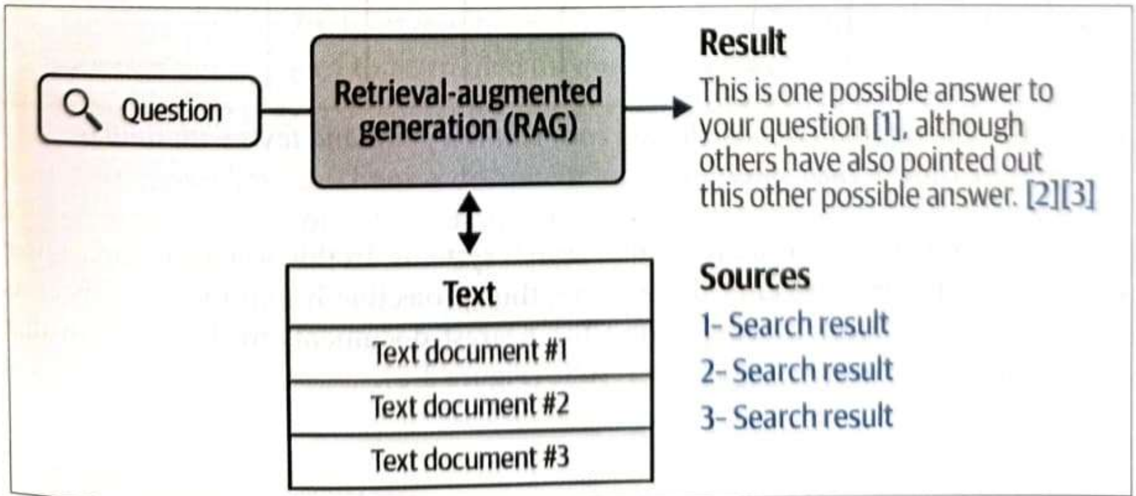
Question
Retrieval-augmented generation (RAG)
Text
Text document #1
Text document #2
Text document #3
Result
This is one possible answer to your question [1], although others have also pointed out this other possible answer. [2][3]
Sources
1- Search result
2- Search result
3- Search result

Figure 8-3. A RAG system formulates an answer to a question and (preferably) cites its information sources.

Dense Retrieval

Recall that embeddings turn text into numeric representations. Those can be thought of as points in space, as we can see in Figure 8-4. Points that are close together mean that the text they represent is similar. So in this example, text 1 and text 2 are more similar to each other (because they are near each other) than text 3 (because it's farther away).

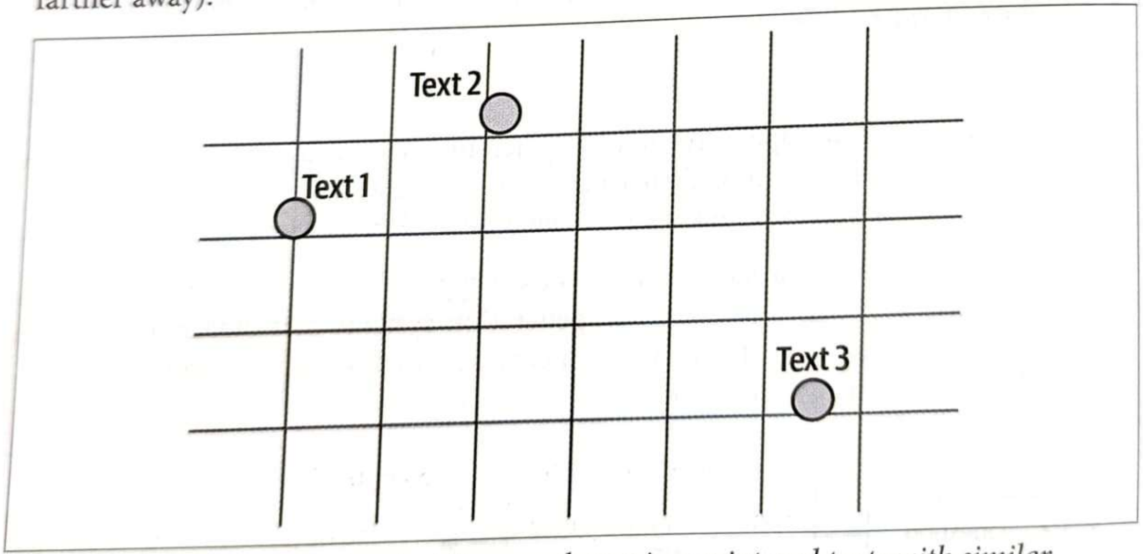

Figure 8-4. The intuition of embeddings: each text is a point and texts with similar meaning are close to each other.

Advanced RAG Techniques

There are several additional techniques to improve the performance of RAG systems. Some of them are laid out here.

Query rewriting

If the RAG system is a chatbot, the preceding simple RAG implementation would likely struggle with the search step if a question is too verbose, or to refer to context in previous messages in the conversation. This is why it's a good idea to use an LLM to rewrite the query into one that aids the retrieval step in getting the right information. An example of this is a message such as:

> User Question: "We have an essay due tomorrow. We have to write about some animal. I love penguins. I could write about them. But I could also write about dolphins. Are they animals? Maybe. Let's do dolphins. Where do they live for example?"

This should actually be rewritten into a query like:

> Query: "Where do dolphins live"

This rewriting behavior can be done through a prompt (or through an API call). Cohere's API, for example, has a dedicated query-rewriting mode for `co.chat`.