Semantic Search & Retreival Augmented Generation

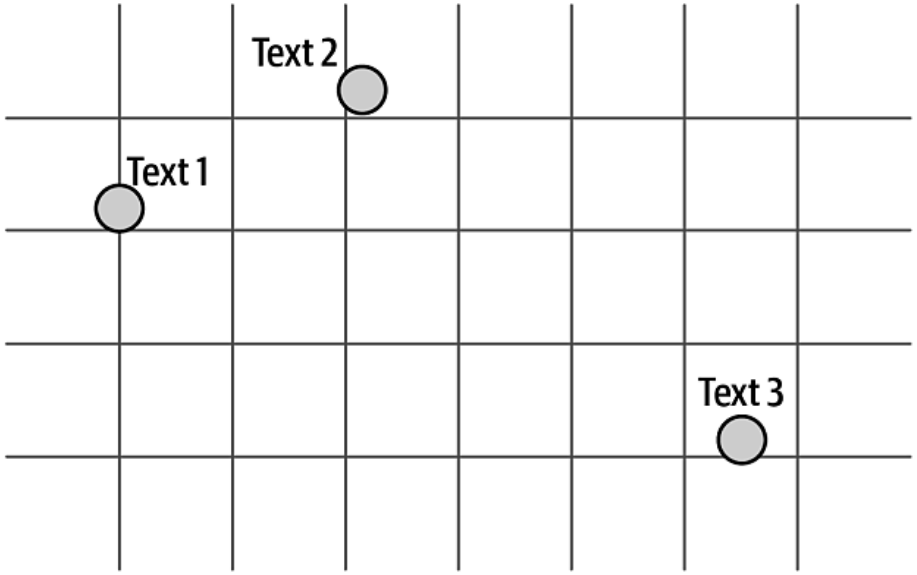
CS XXX: Introduction to Large Language Models

Contents

- Embeddings
- Semantic Search
- Re-ranking after search
- Retrieval Evaluation Metrics: MAP
- Retrieval Augmented Generation (RAG)
- RAG Evaluation

Embeddings

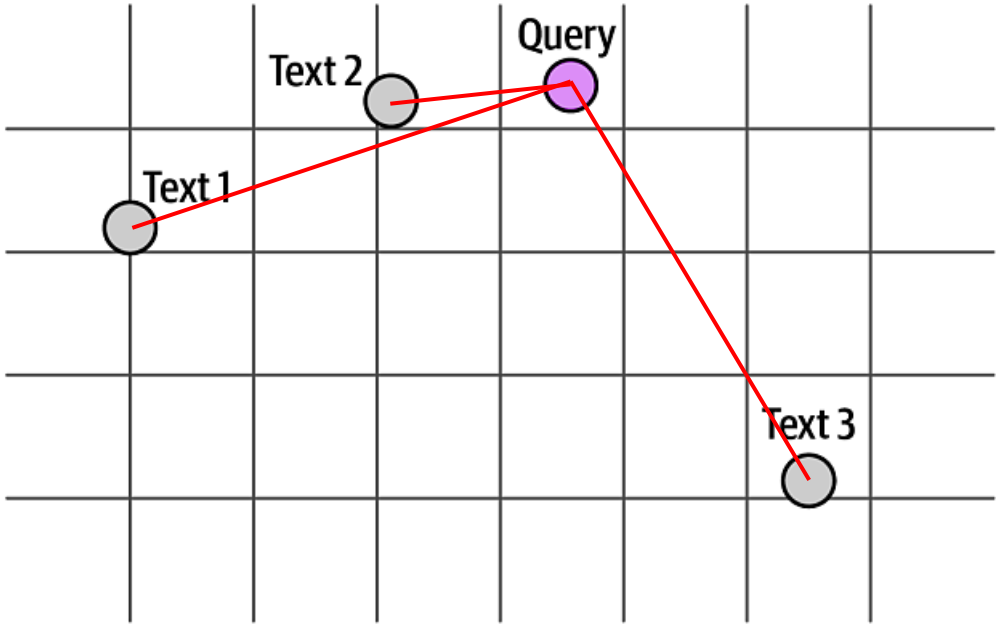
- RECALL
- Recall that embeddings turn text into numeric representations. Those can be thought of as points in space



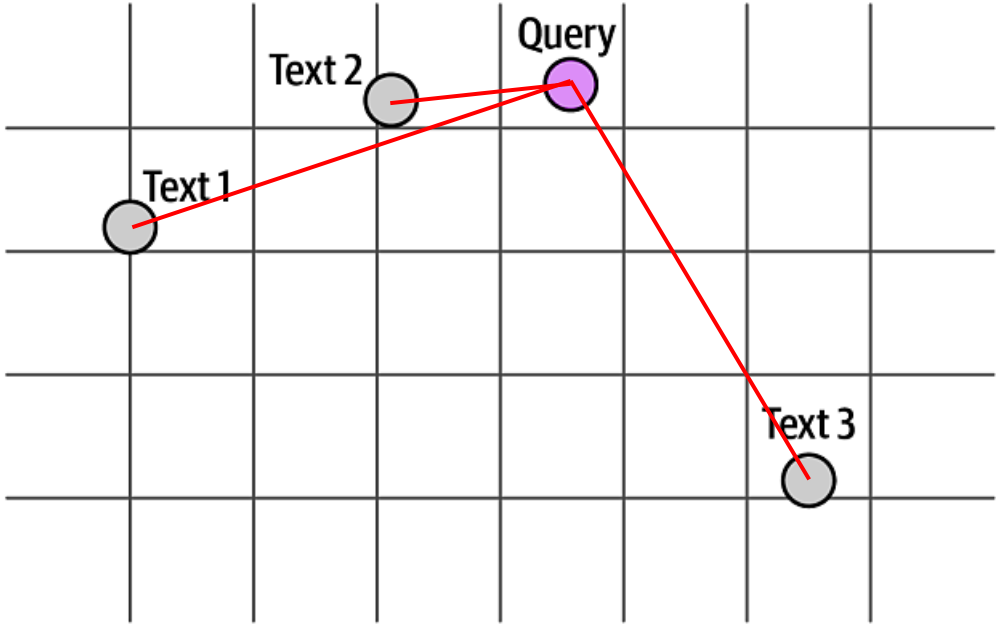
- Points that are close together mean that the text they represent is similar. So in this example, text 1 and text 2 are more similar to each other (because they are near each other) than text 3 (because it's farther away).

Semantic Search

- Points that are close together mean that the text they represent is similar.
- This property is used to build search systems.
- When a user enters a search query, we embed the query, thus projecting it into the same space as our text archive. Then we simply find the nearest documents to the query in that space, and those would be the search results



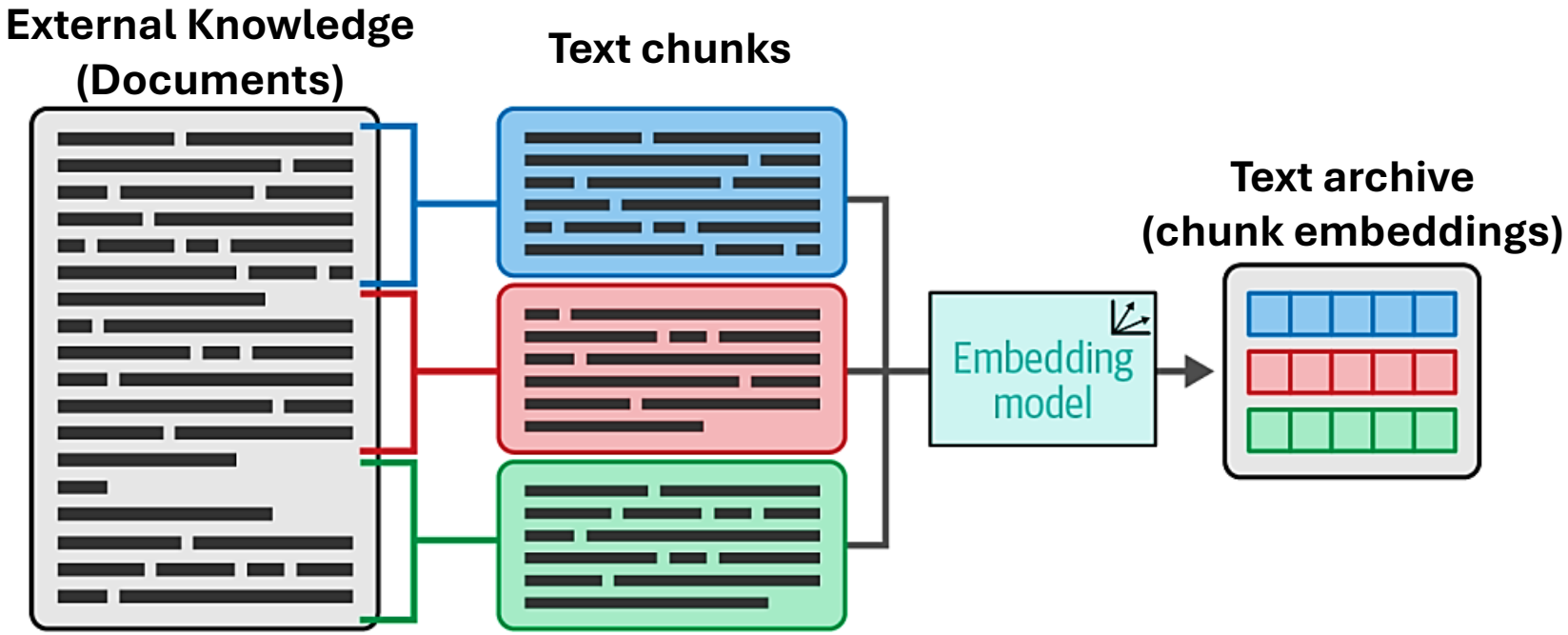
Semantic Search



- Judging by the distances, **Text 2** is the best result for this query.
- We may choose to get the top k most similar results. For example, if $k = 2$ then **Text 2**, and **Text 1** will both be results for this query.

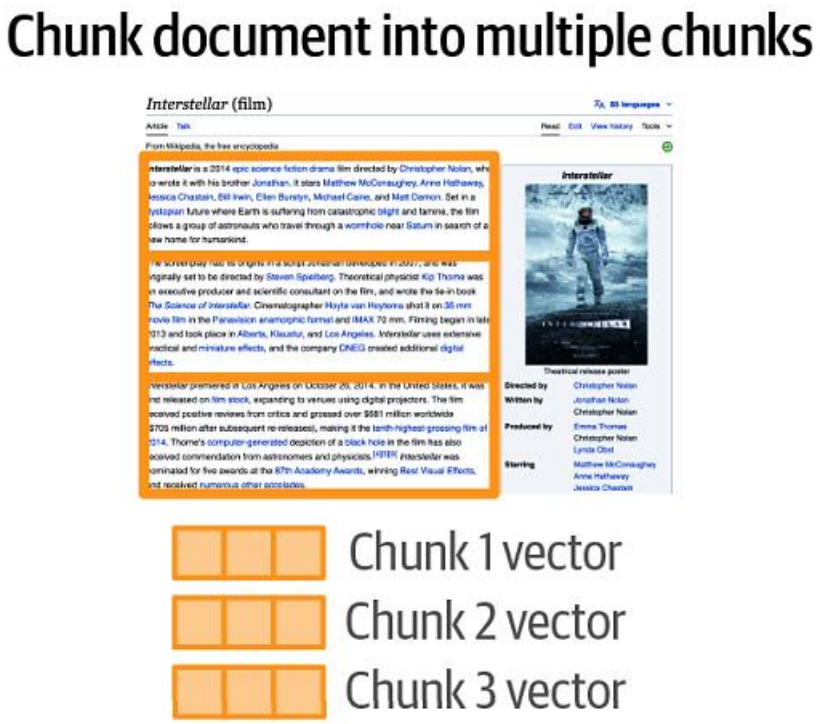
Semantic Search

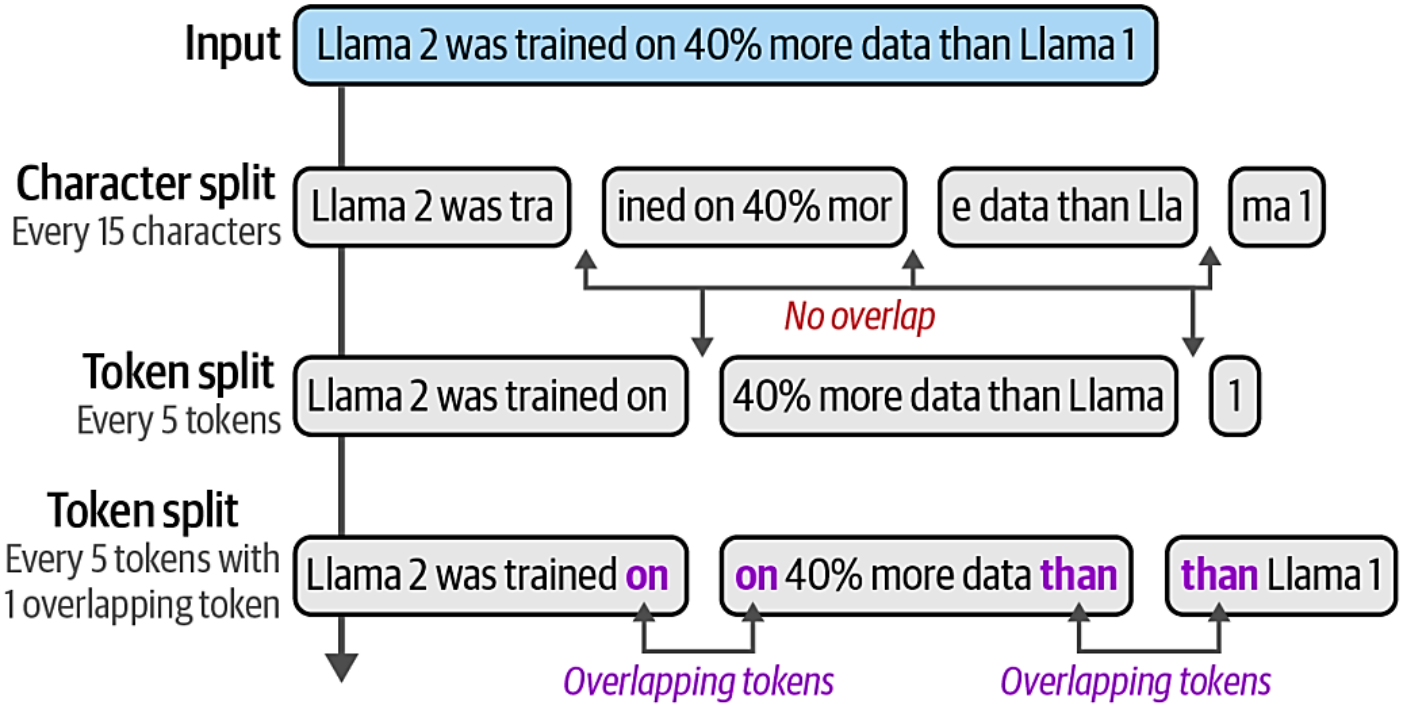
- A large corpus of documents is broken down into small pieces of text called text chunks. Embeddings are created for each chunk to form the text archive embedding space.



Semantic Search

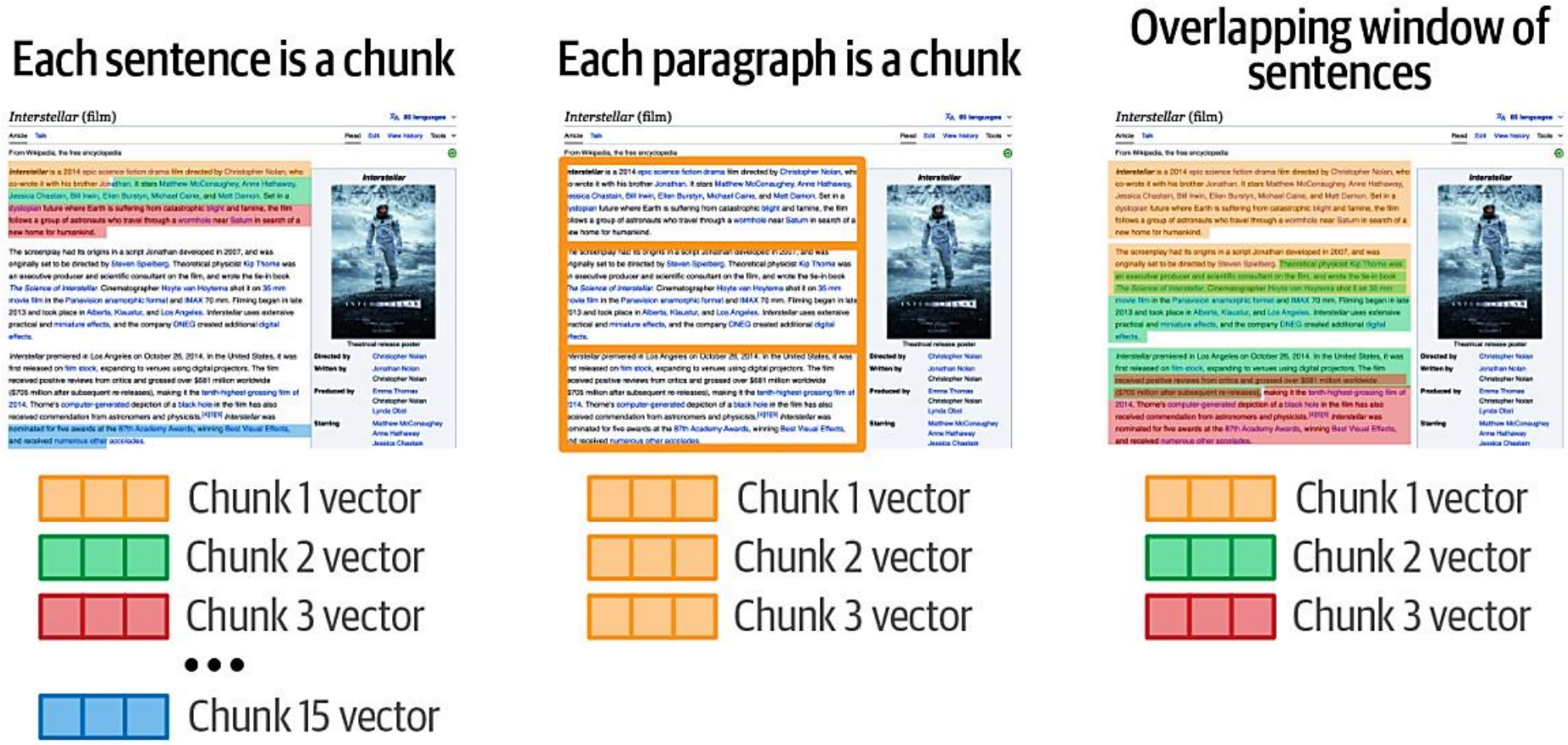
- We chunk the documents into smaller pieces, and embed those chunks.





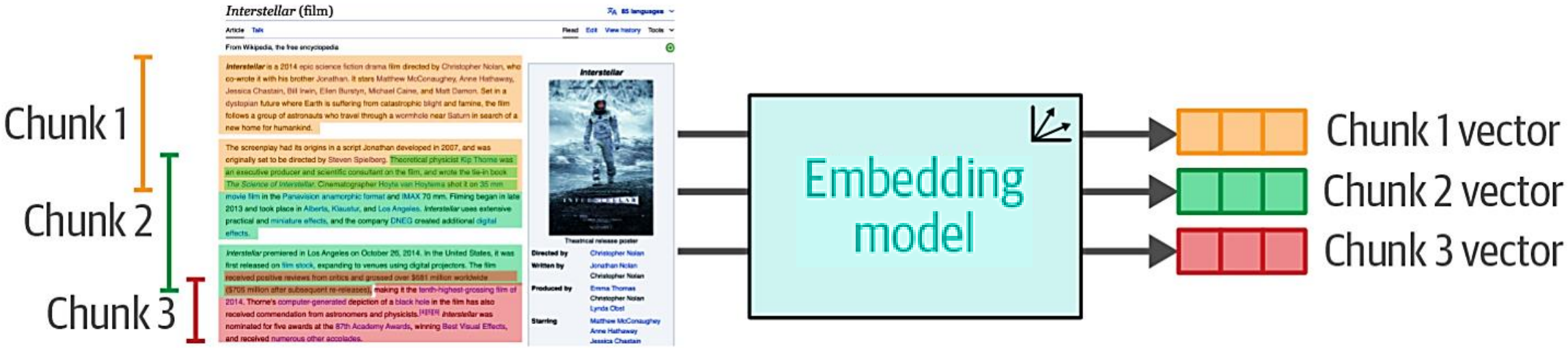
Semantic Search

- The best way of chunking a long text will depend on the types of texts and queries your system anticipates.



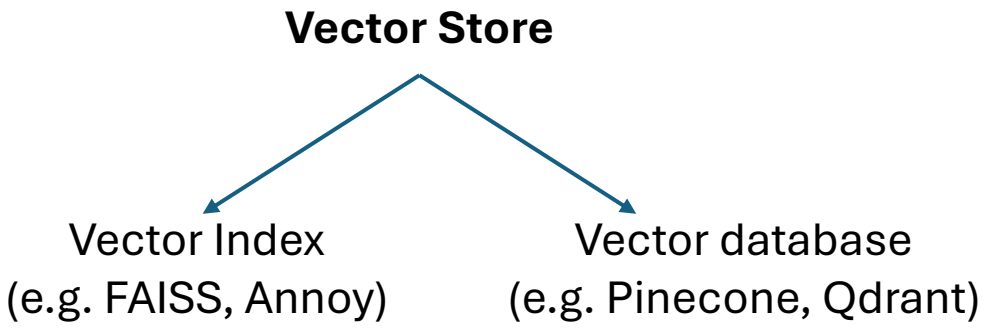
Semantic Search

- Chunking the text into overlapping segments is one strategy to retain more of the context around different segments.
- Chunks derive a lot of their meaning from the text around them. Overlapping chunks allows the inclusion of surrounding text that also appears in adjacent chunks.

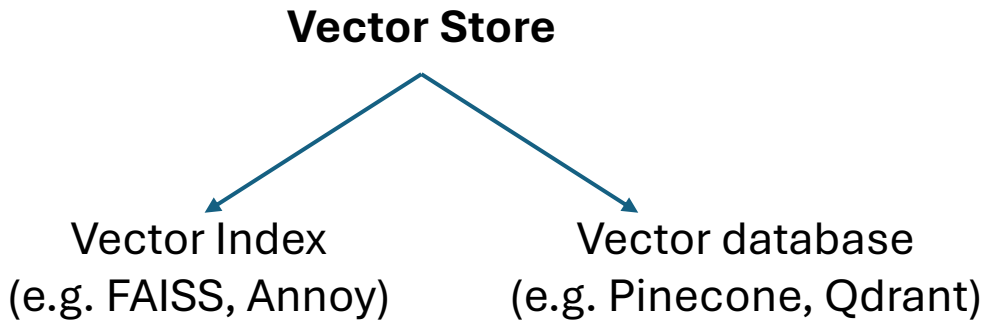


Semantic Search

- Once the query is embedded, we need to find the nearest vectors (most similar chunks) to it from our text archive.
- A *vector store* (vector index or database) is used to store the chunk embeddings. These vector stores are optimized to quickly retrieve similar embeddings to the query even if we have a very large number of points.



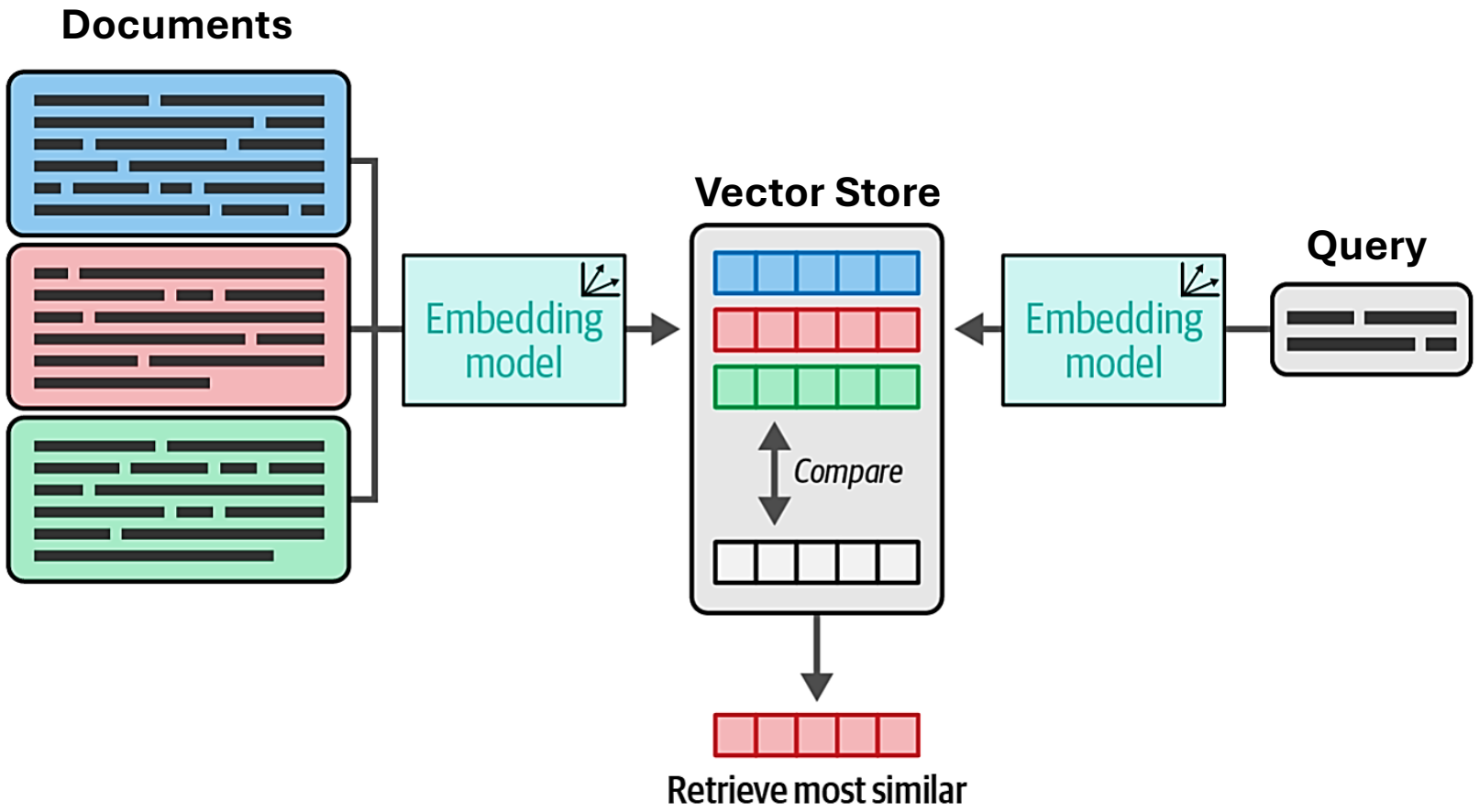
Semantic Search



- A vector index is a data structure designed to organize and facilitate efficient retrieval of high-dimensional vector data.
- A vector database is a comprehensive system that uses vector indices internally to offer scalable, persistent, and metadata-enriched functionality for managing vector data. A vector database allows you to add or delete vectors without having to rebuild the index.

Semantic Search

- We can now search the dataset using any query we want. We simply embed the query and present its embedding to the index, which will retrieve the most similar sentence



Semantic Search

- We can now search the dataset using any query we want. We simply embed the query and present its embedding to the index, which will retrieve the most similar sentence

```
Query: 'how precise was the science'
Nearest neighbors:
```

	texts	distance
0	It has also received praise from many astronomers for its scientific accuracy and portrayal of theoretical astrophysics	10757.379883
1	Caltech theoretical physicist and 2017 Nobel laureate in Physics[4] Kip Thorne was an executive producer, acted as a scientific consultant, and wrote a tie-in book, The Science of Interstellar	11566.131836
2	Interstellar uses extensive practical and miniature effects and the company Double Negative created additional digital effects	11922.833008

Semantic Search

- What happens, for example, if the texts don't contain the answer? We still get results and their distances. For example:

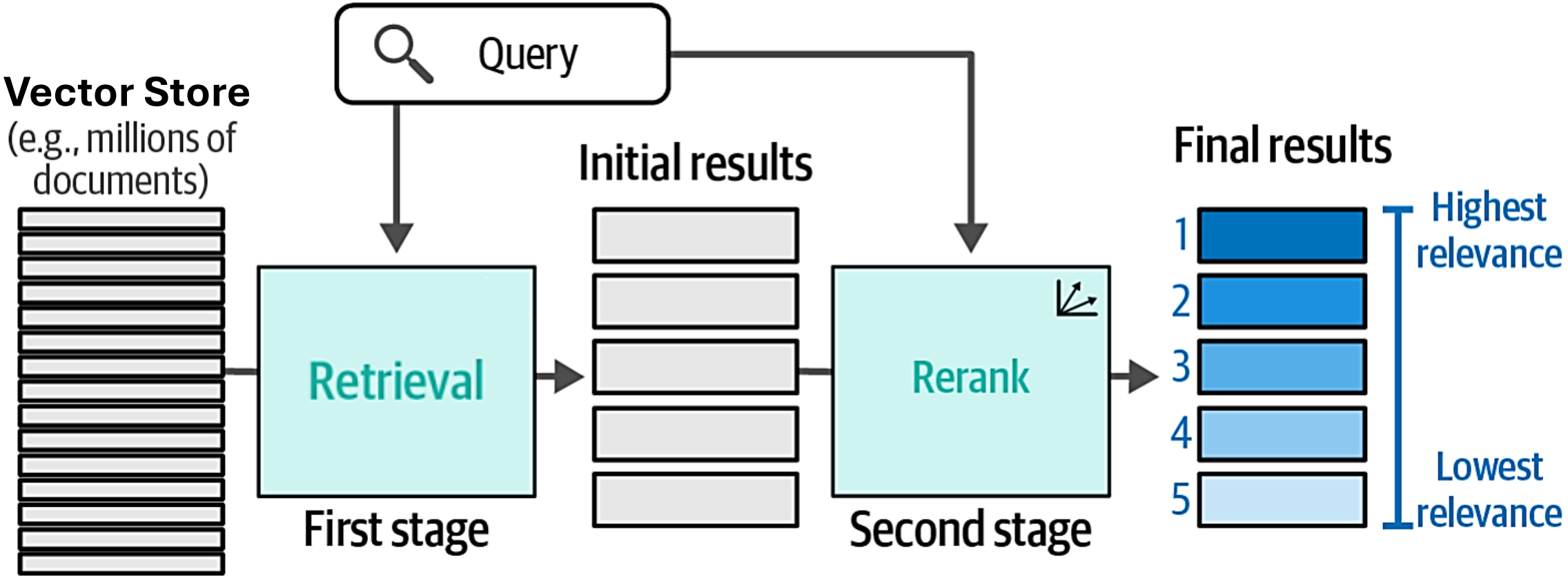
```
Query:'What is the mass of the moon?'
Nearest neighbors:
```

	texts	distance
0	The film had a worldwide gross over $677 million (and $773 million with subsequent re-releases), making it the tenth-highest grossing film of 2014	1.298275
1	It has also received praise from many astronomers for its scientific accuracy and portrayal of theoretical astrophysics	1.324389
2	Cinematographer Hoyte van Hoytema shot it on 35 mm movie film in the Panavision anamorphic format and IMAX 70 mm	1.328375

- In cases like this, one possible heuristic is to set a threshold level—a maximum distance for relevance e.g. only distance <= 0.9 is relevant.

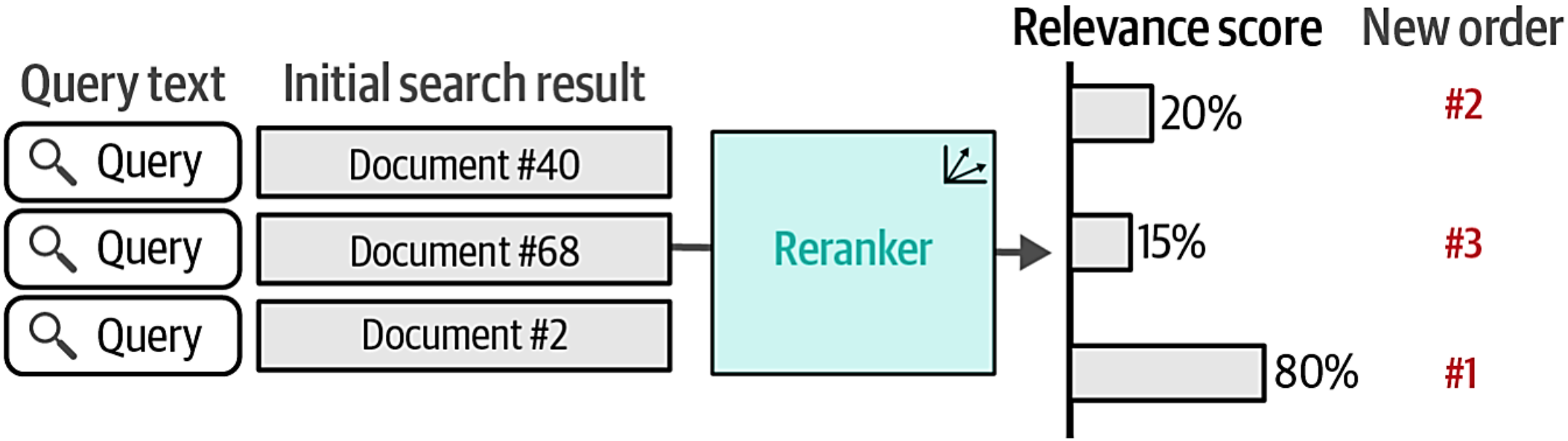
Re-ranking after search

- The re-ranking step includes changing the order of changing the order of the search results based on relevance to the search query. This one step can vastly improve search results.



Re-ranking after search

- A reranker assigns a relevance score to each document by comparing the document and the query.



Re-ranking after search

- A reranker assigns a relevance score to each document by comparing the document and the query.

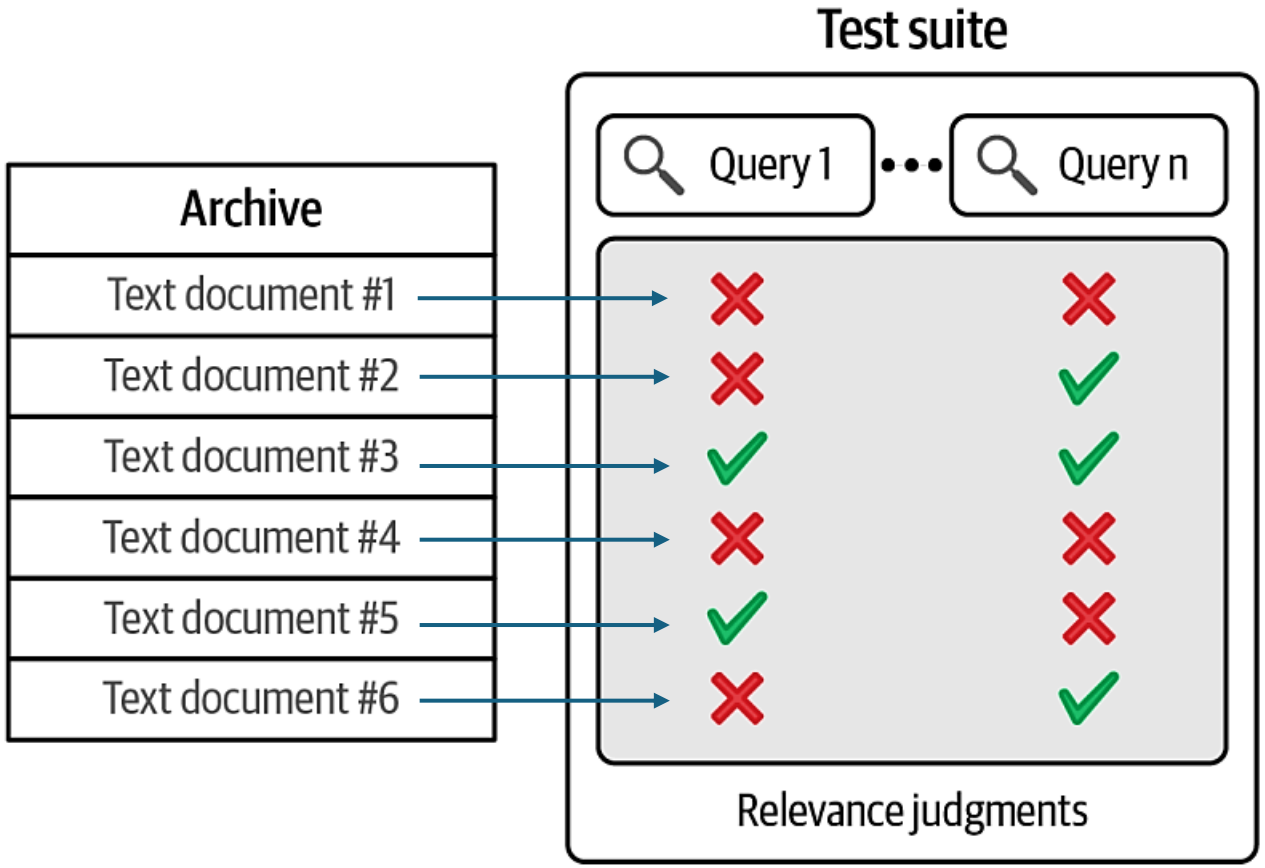
```
      Query: 'how precise was the science'

0 0.1698185 It has also received praise from many astronomers for its scien-
tific accuracy and portrayal of theoretical astrophysics
1 0.07004896 The film had a worldwide gross over $677 million (and $773 mil-
lion with subsequent re-releases), making it the tenth-highest grossing film
of 2014
2 0.0043994132 Caltech theoretical physicist and 2017 Nobel laureate in Phys-
ics[4] Kip Thorne was an executive producer, acted as a scientific consultant,
and wrote a tie-in book, The Science of Interstellar
```

- This shows the reranker is much more confident about the first result, assigning it a relevance score of 0.16, while the other results are scored much lower in relevance.

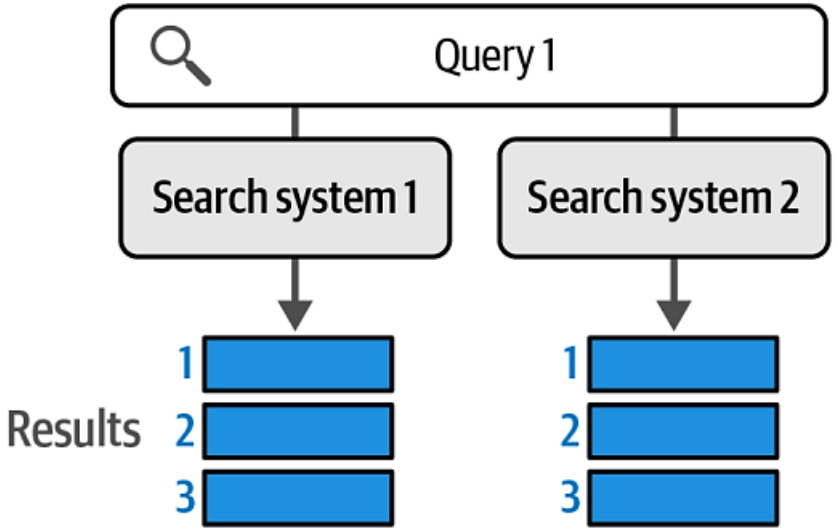
Retrieval Evaluation Metrics

- To evaluate search systems, we need a test suite including queries and relevance judgments indicating which documents in our archive are relevant for each query



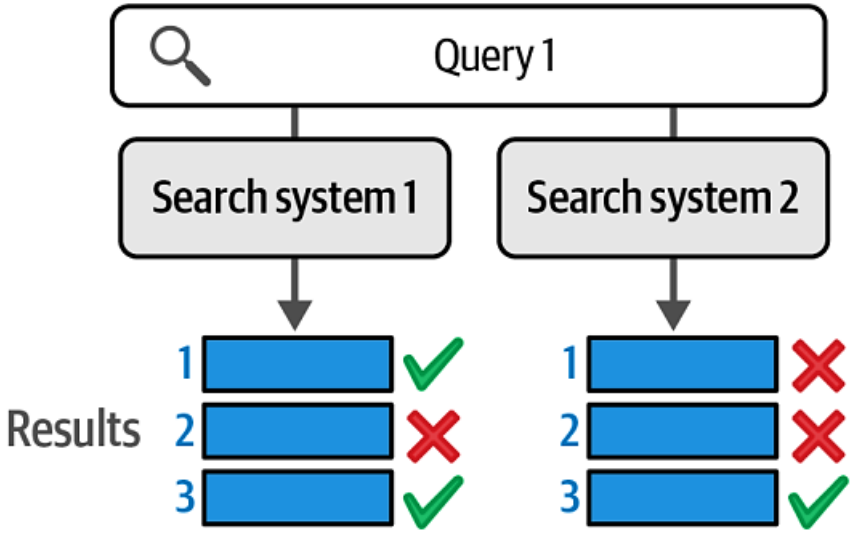
Retrieval Evaluation Metrics

- Using the test suite, we can proceed to explore evaluating search systems. Let's assume we pass query 1 to two different search systems.



Retrieval Evaluation Metrics

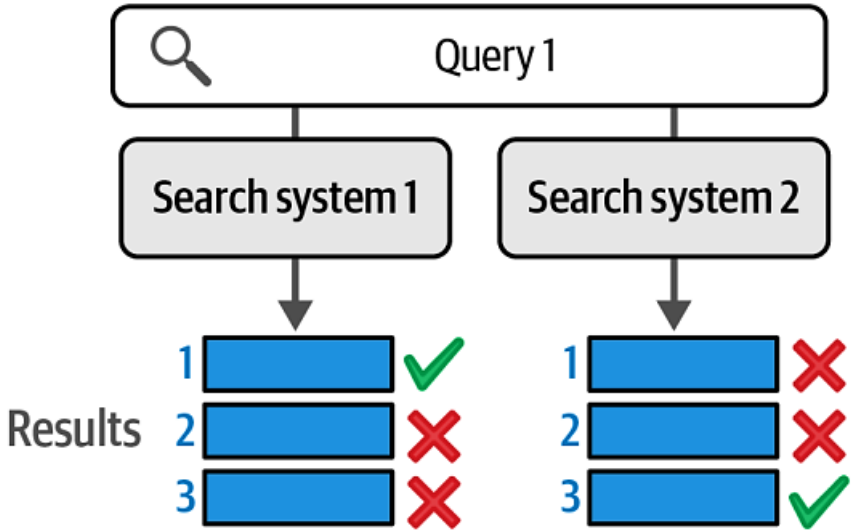
- To compare two search systems, we pass the same query from our test suite to both systems and look at their top results.



- Looking at the relevance judgments from our test suite, we can see that system 1 did a better job than system 2. This shows us a clear case where system 1 is better than system 2.

Retrieval Evaluation Metrics

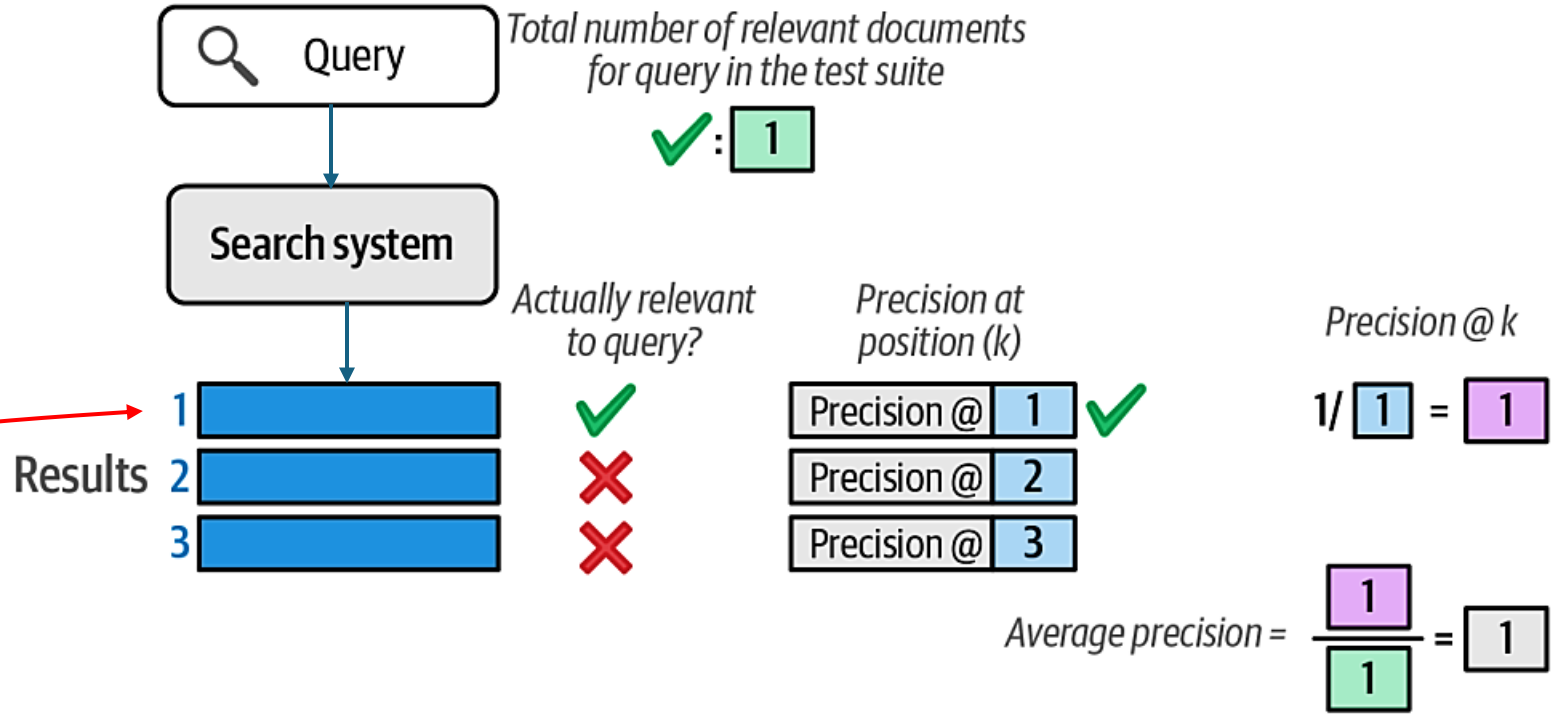
- But what about a case where both systems only get one relevant result out of three, but they're in different positions?



- We need a scoring system that rewards system 1 for assigning a high position to a relevant result—even though both systems retrieved only one relevant result in their top three results.

Retrieval Evaluation Metrics

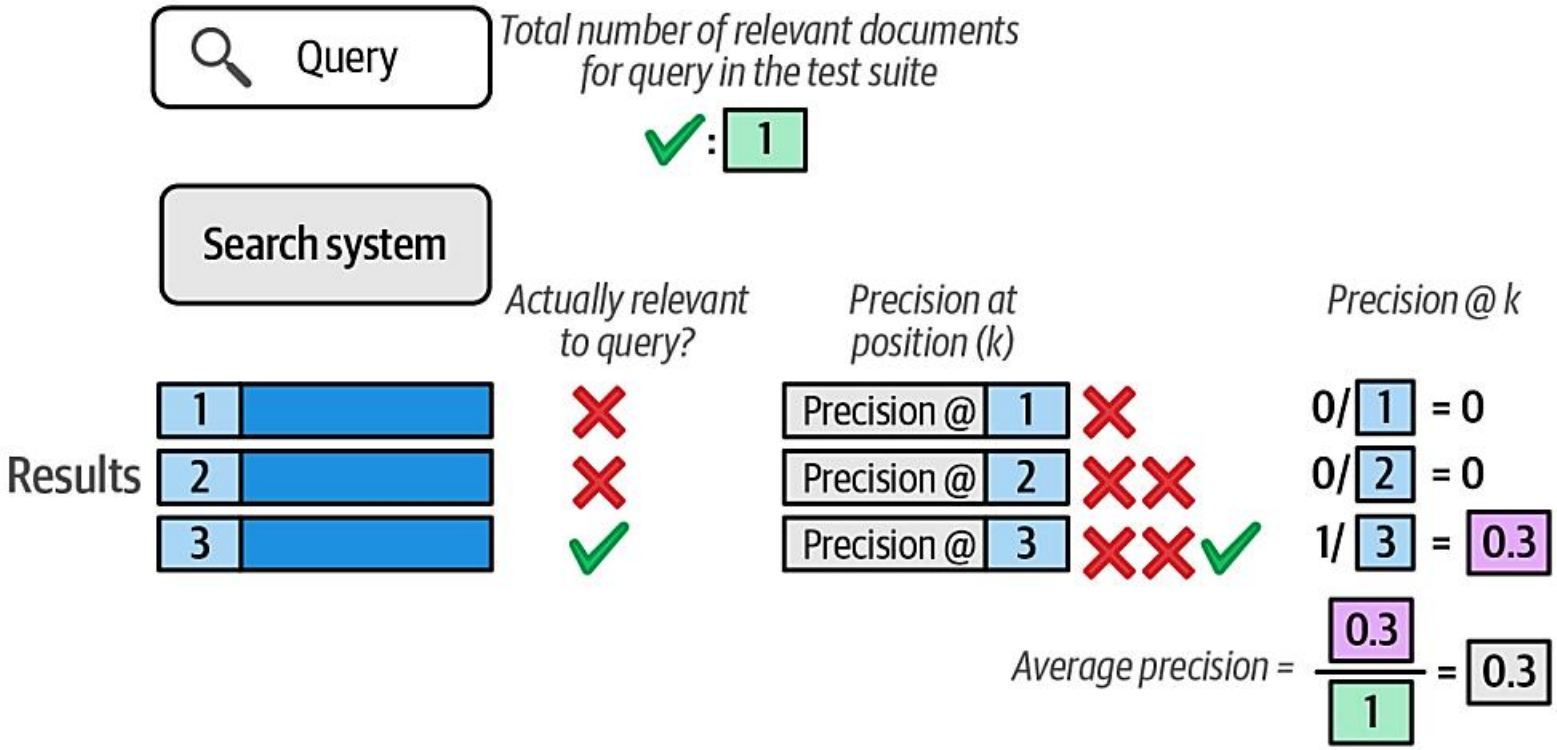
- One common way to assign numeric scores in this scenario is average precision. We calculate precision at each position, starting at position 1



- the search system placed the relevant result (the only available one for this query) at the top.

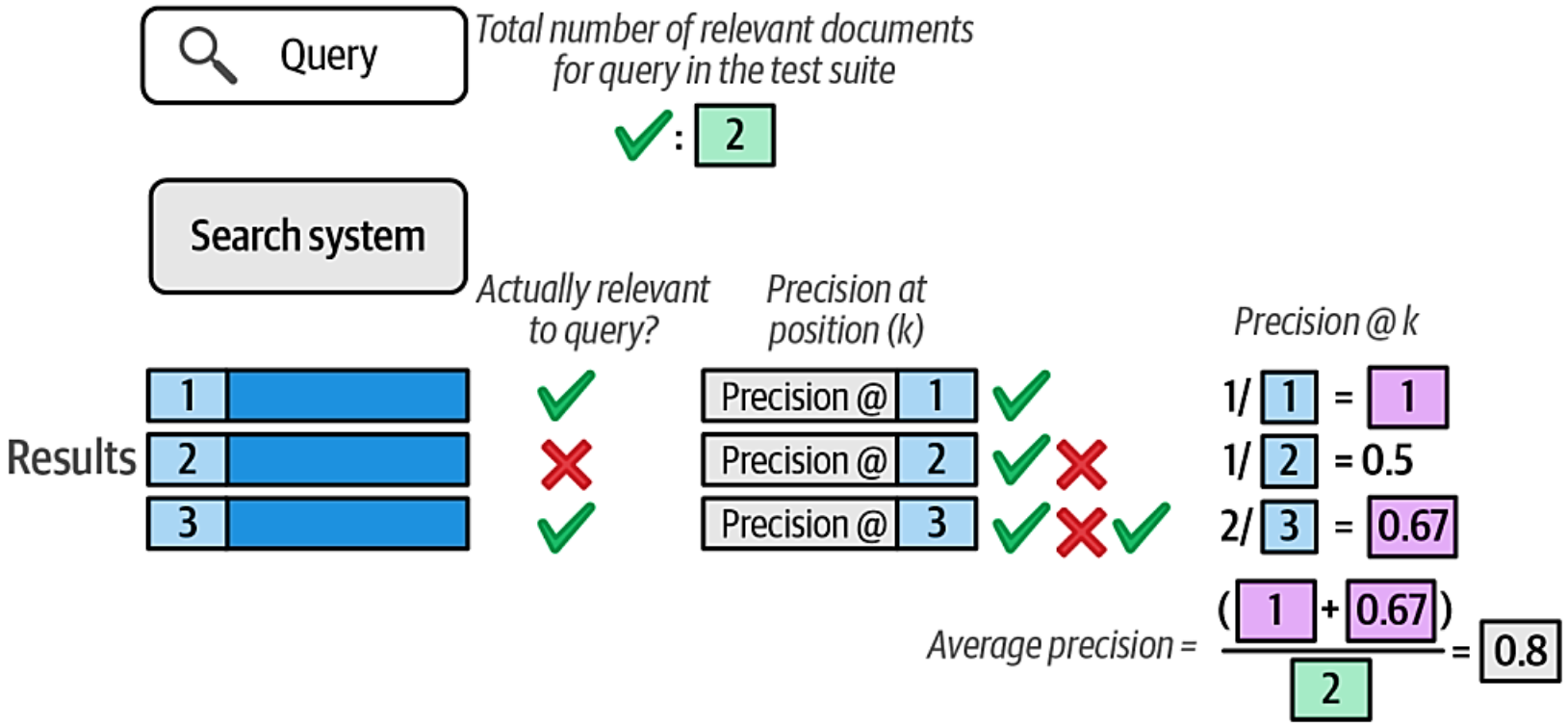
Retrieval Evaluation Metrics

- What if the system actually placed the only relevant result at the third position, however?
- If the system places nonrelevant documents ahead of a relevant document, its precision score is penalized.



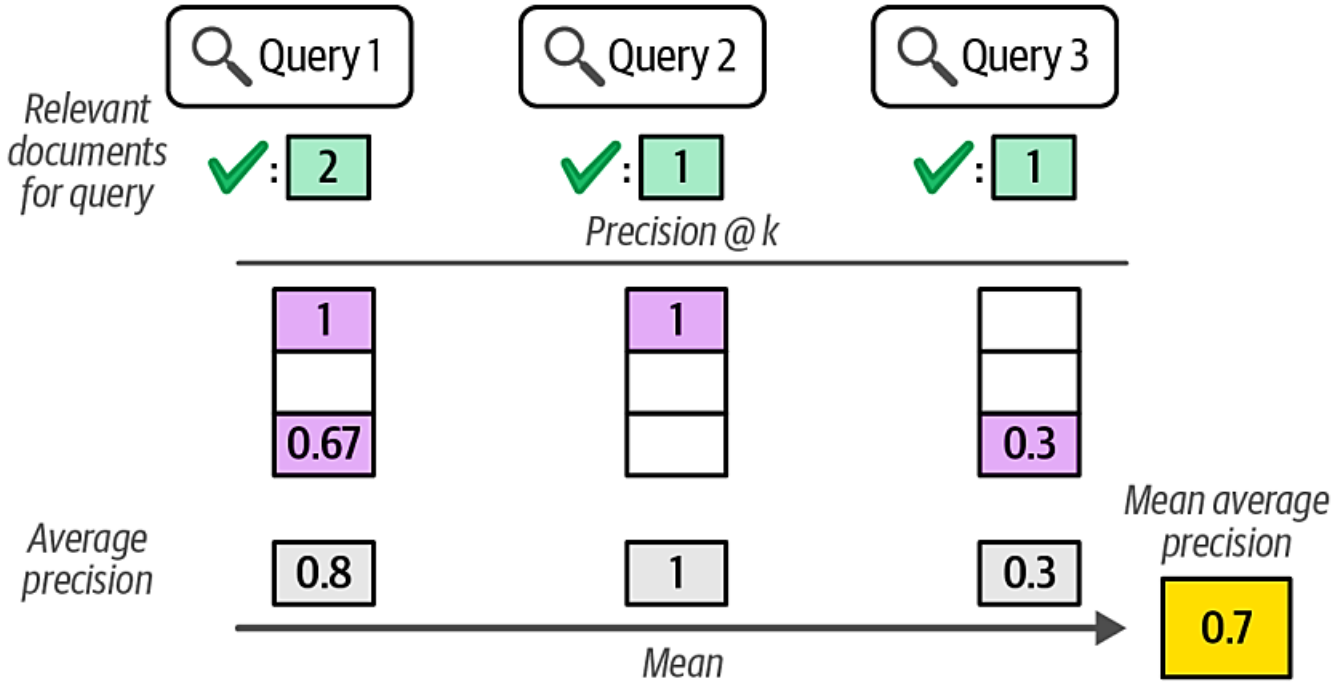
Retrieval Evaluation Metrics

- Now look at a query with more than one relevant document.
- Average precision of a document with multiple relevant documents considers the precision at *k* results of all the relevant documents



Retrieval Evaluation Metrics

- We can extend this knowledge to a metric that can score a search system against all the queries in our test suite.
- The mean average precision takes the mean of the average precision scores of a system for every query in the test suite.



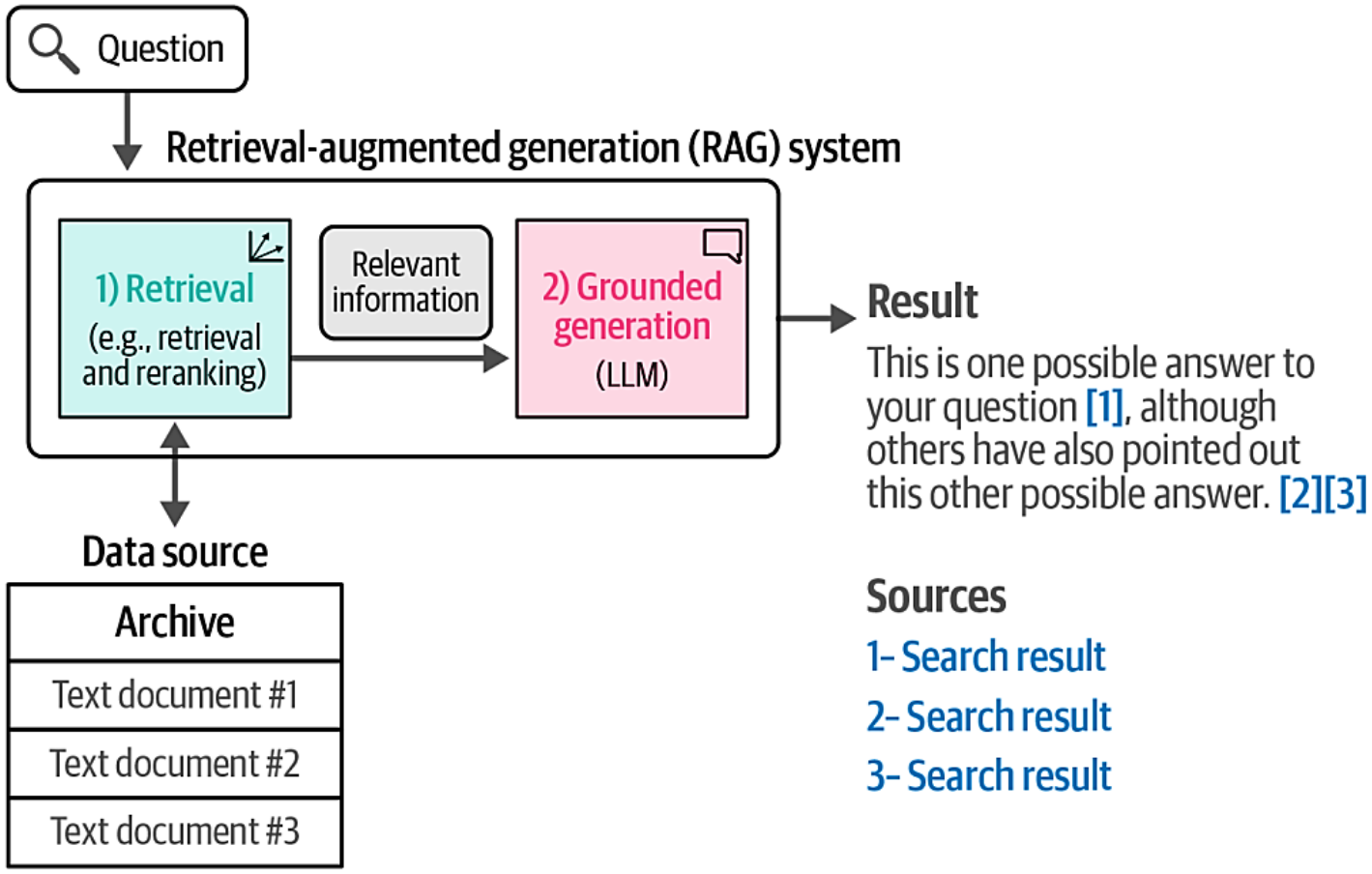
Retrieval Augmented Generation

- Retrieval Augmented Generation (RAG) systems incorporate search capabilities in addition to generation capabilities. They can be seen as an improvement to generation systems because they reduce their hallucinations and improve their factuality.

- In RAG systems we add an LLM to the end of the search pipeline. We present the question and the top retrieved documents to the LLM, and ask it to answer the question given the context provided by the search results. This generation step is called “grounded generation” because the retrieved relevant information we provide the LLM establishes a certain context that grounds the LLM in the domain we’re interested in.

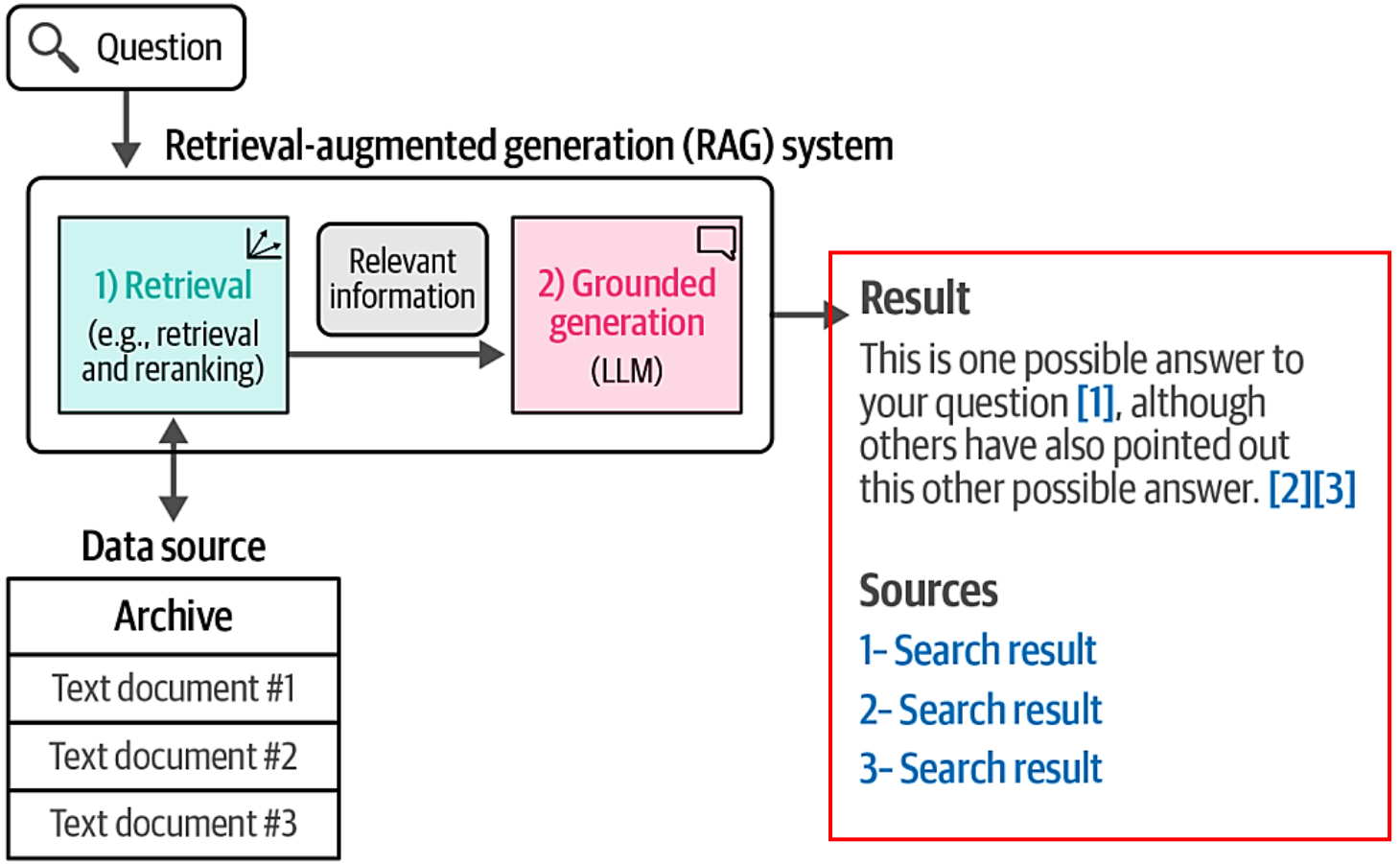
Retrieval Augmented Generation

- In RAG systems we add an LLM to the end of the search pipeline. We present the question and the top retrieved documents to the LLM, and ask it to answer the question given the context provided by the search results.



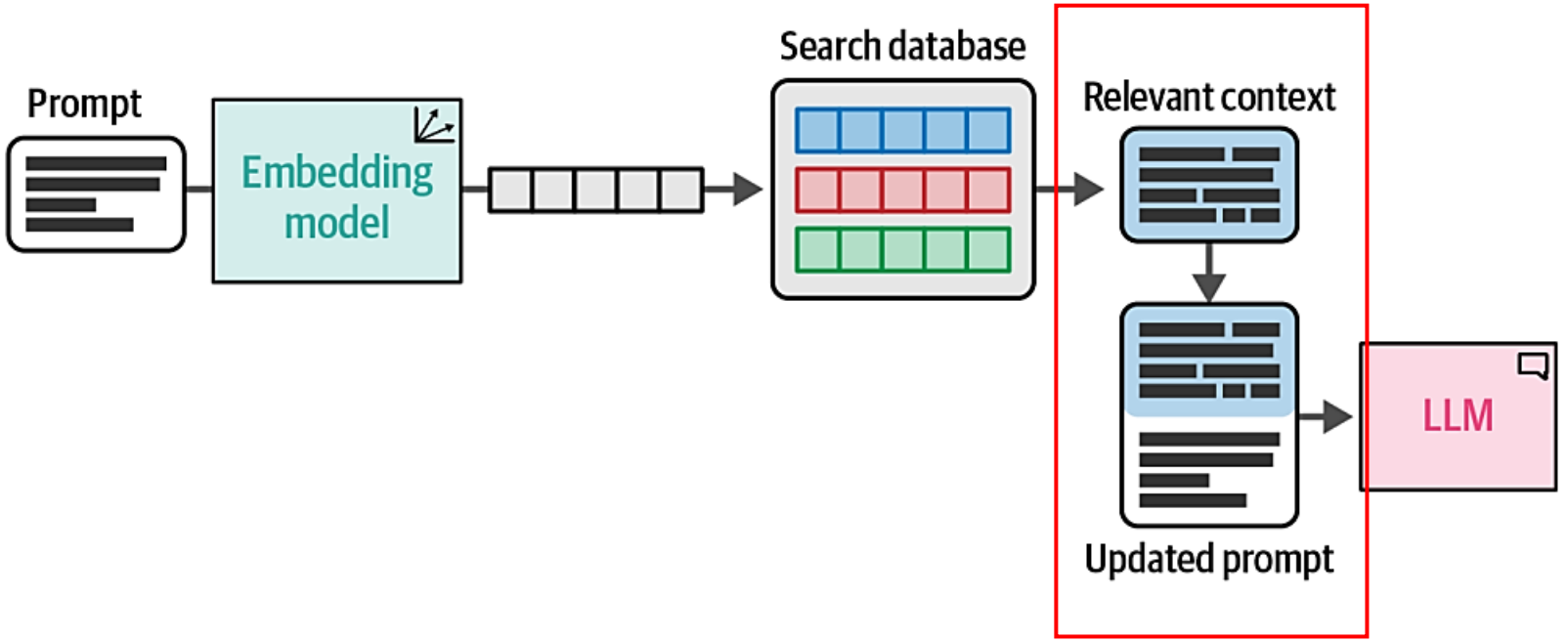
Retrieval Augmented Generation

- In RAG systems we add an LLM to the end of the search pipeline. We present the question and the top retrieved documents to the LLM, and ask it to answer the question given the context provided by the search results.



Generative search formulates answers and summaries at the end of a search pipeline while citing its sources (returned by the previous steps in the search system).

Retrieval Augmented Generation



- A prompt template plays a vital part in the RAG pipeline. It is the central place where we communicate the relevant documents to the LLM. To do so, we will create an additional input variable named `context` that can provide the LLM with the retrieved documents.

Retrieval Augmented Generation

- A prompt template plays a vital part in the RAG pipeline. It is the central place where we communicate the relevant documents to the LLM. To do so, we will create an additional input variable named `context` that can provide the LLM with the retrieved documents.

```
template = """<|user|>
Relevant information:
{context}

Provide a concise answer the following question using the relevant information
provided above:
{question}<|end|>
<|assistant|>"""
```

RAG Evaluation

- There are still ongoing developments in how to evaluate RAG models. While human evaluation is always preferred, there are approaches that attempt to automate these evaluations by having a capable LLM act as a judge (called LLM-as-a-judge) and score the different generations along the different axes. Ragas is a software library that does exactly this. It also scores some additional useful metrics like:
  - Faithfulness: Whether the answer is consistent with the provided context
  - Answer relevance: How relevant the answer is to the question

References

- Alammar, J., & Grootendorst, M. Hands-On Large Language Models: Language Understanding and Generation. O'Reilly Media.