Improve your RAG pipelines with semantic re-ranking

Susan Shu Chang

Data Day Texas 2025

About the speaker

Principal data scientist at Elastic (Elasticsearch)

About the speaker

Principal data scientist at Elastic (Elasticsearch)

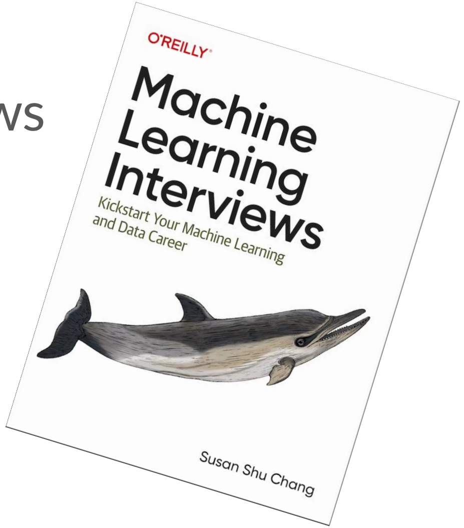
6x PyCon speaker 🌎

About the speaker

Principal data scientist at Elastic (Elasticsearch)

6x PyCon speaker 🌎

Author, Machine Learning Interviews



👋Who travelled more than 3 hours to get here?

👋Who travelled less than 3 hours to get here?

Overview

1. Very quick primer on RAG
2. Search and recommendations: A story
3. Improving Retrieval: The “R” in RAG
4. 🥫 Secret sauce?! How rerankers are trained
5. All together: Rerankers in RAG systems

1

Very quick primer on RAG

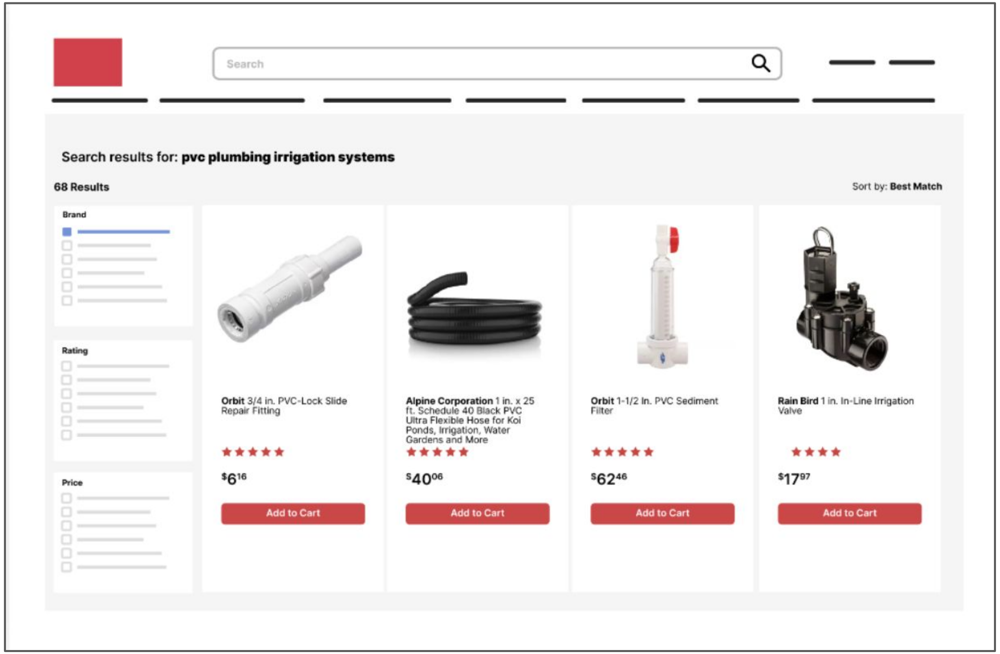
British Columbia

Air Canada found liable for chatbot's bad advice on plane tickets

Airline's claim that online helper was responsible for its own actions was 'remarkable': small claims court

Jason Proctor · CBC News · Posted: Feb 15, 2024 3:38 PM EST | Last Updated: February 16, 2024

RAG has become a standard to avoid hallucinations



What material list and tools do I need to build an irrigation system for my 1 acre back yard in Detroit, MI?

Accurate information enhances user experience

Image source: Oxana Melis, via Unsplash

RAG has become a standard to avoid hallucinations

Ideal behaviors: **Do not hallucinate or display**

- Products that we don’t have
- Incorrect product descriptions
- Incorrect production functionality

RAG is relatively **low-hanging fruit** to provide GenAI with accurate info

Retrieve the **most relevant** [support articles / product pages / discount policies] etc…

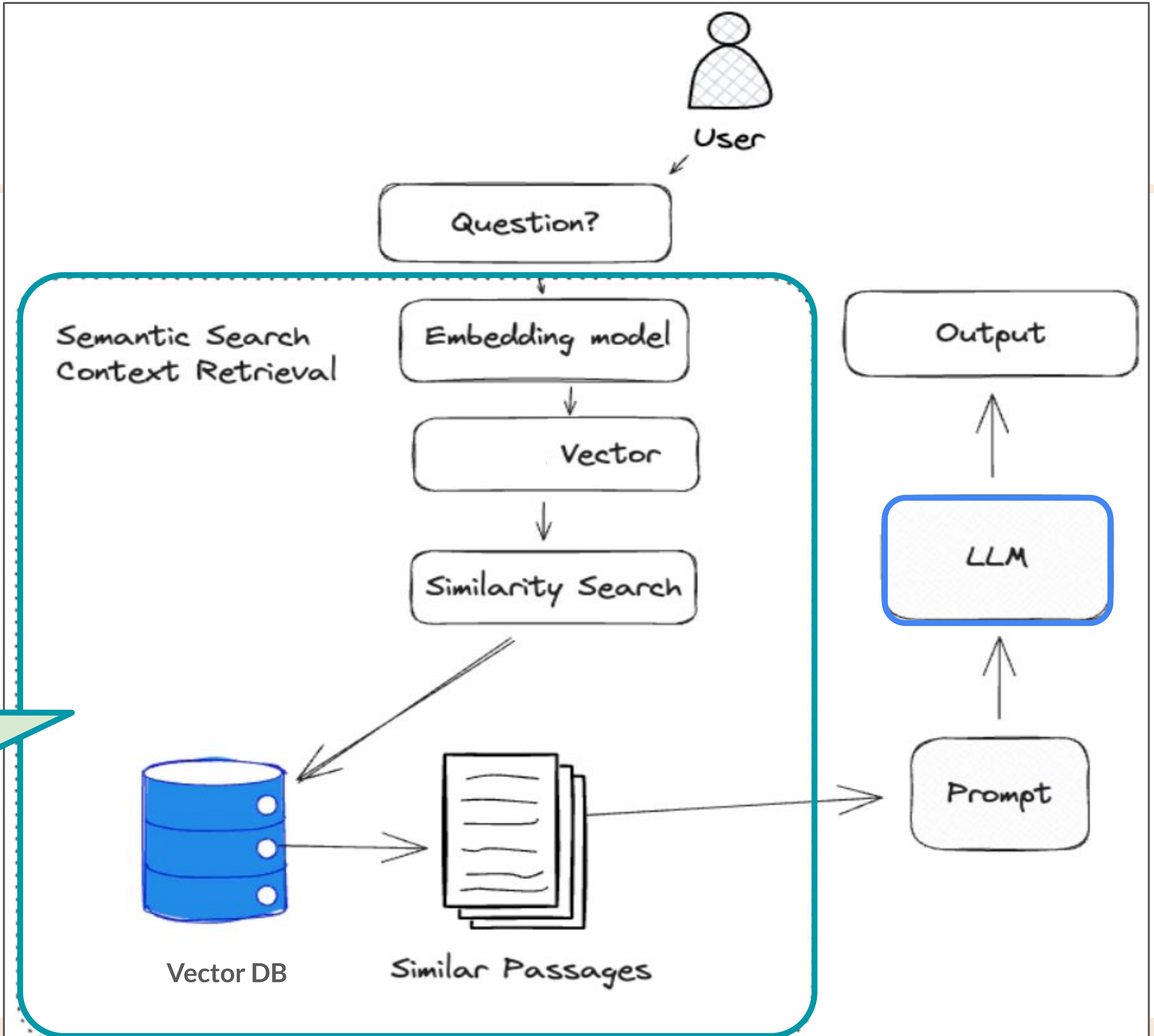
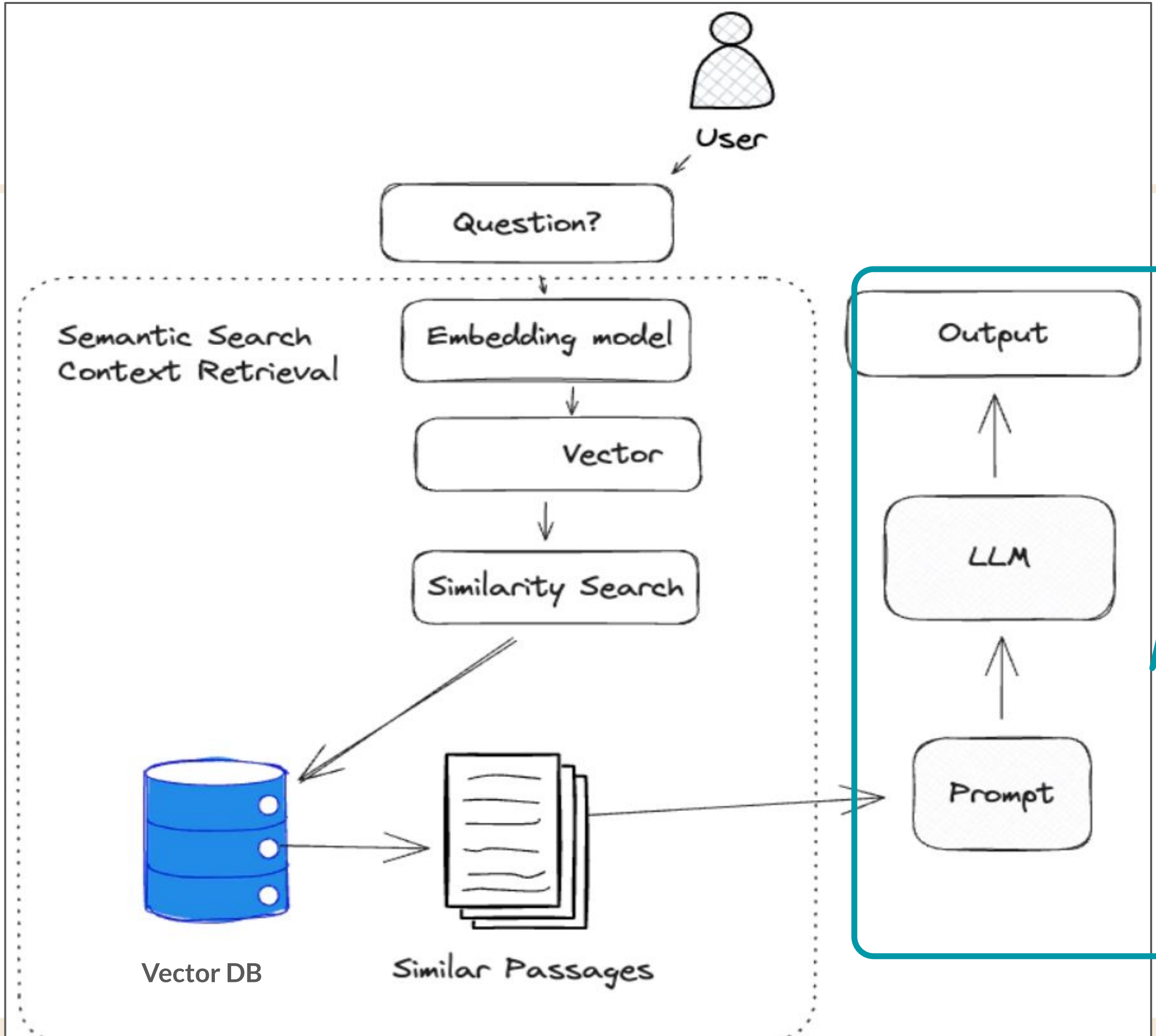


Image: Elastic

RAG is relatively **low-hanging fruit** to provide GenAI with accurate info

...But you need to retrieve the right results



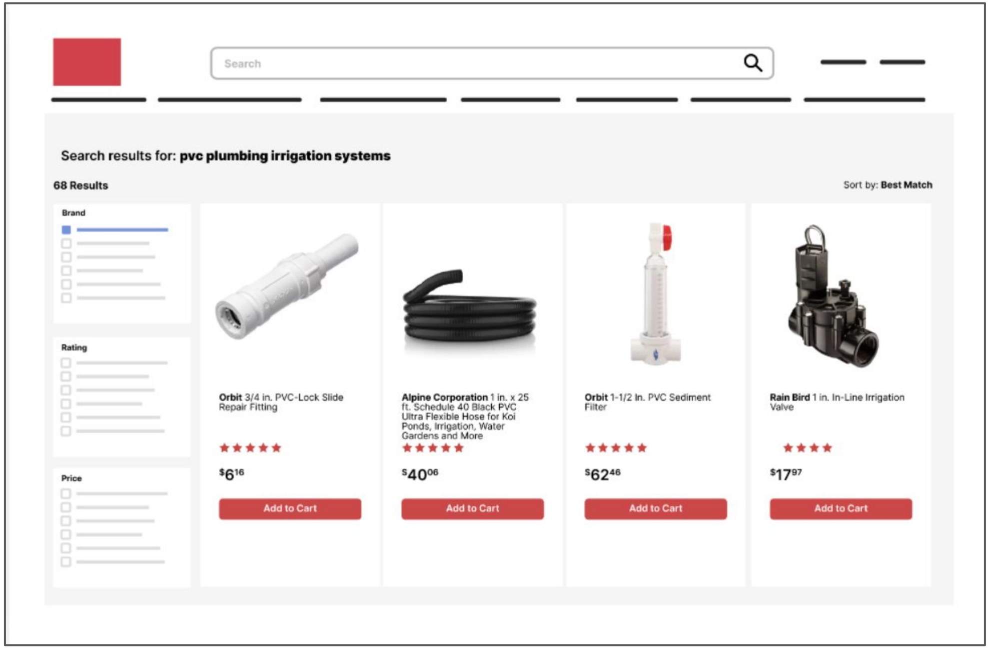
Pass retrieved info to LLM/ Chatbot to **generate** the final output

Image: Elastic

2

Search and recommendations: A story

Most search functions have at least *keyword based* search



BM25 is a common lexical/keyword based function

Many ways of tuning keyword search

Query rules

roof gutter

roofgutter

foof guter

Type	Match Requirements
`exact`	Rule metadata matches the specified value exactly.
`fuzzy`	Rule metadata matches the specified value within an allowed Levenshtein edit distance.
`prefix`	Rule metadata starts with the specified value.
`suffix`	Rule metadata ends with the specified value.
`contains`	Rule metadata contains the specified value.
`lt`	Rule metadata is less than the specified value.
`lte`	Rule metadata is less than or equal to the specified value.

...and much more

Source: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-using-query-rules.html#query-rule-criteria

Bag-of-words / keyword based representations can't account for semantics

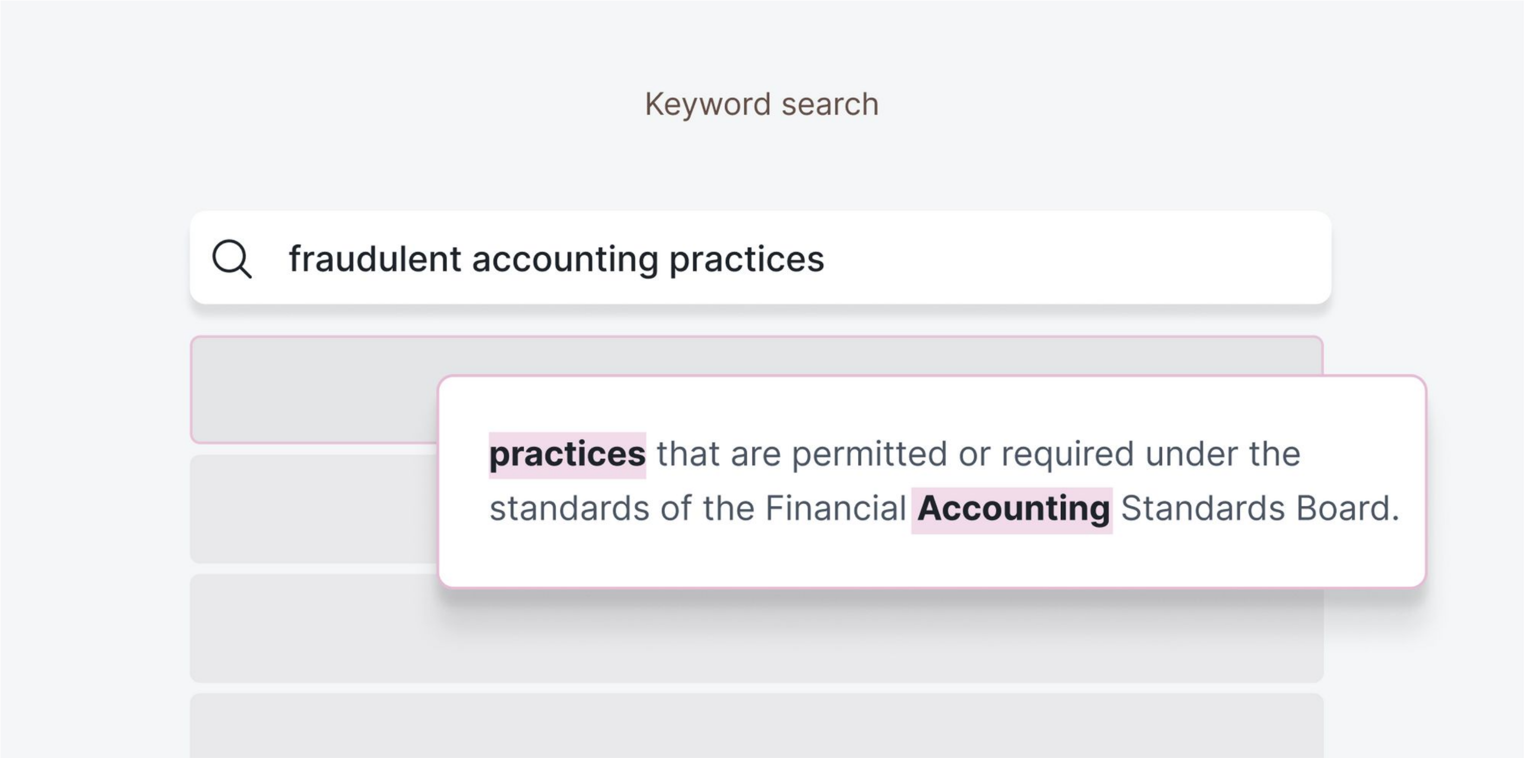


Image: DeepJudge

Language tasks evolve to understand context/semantics

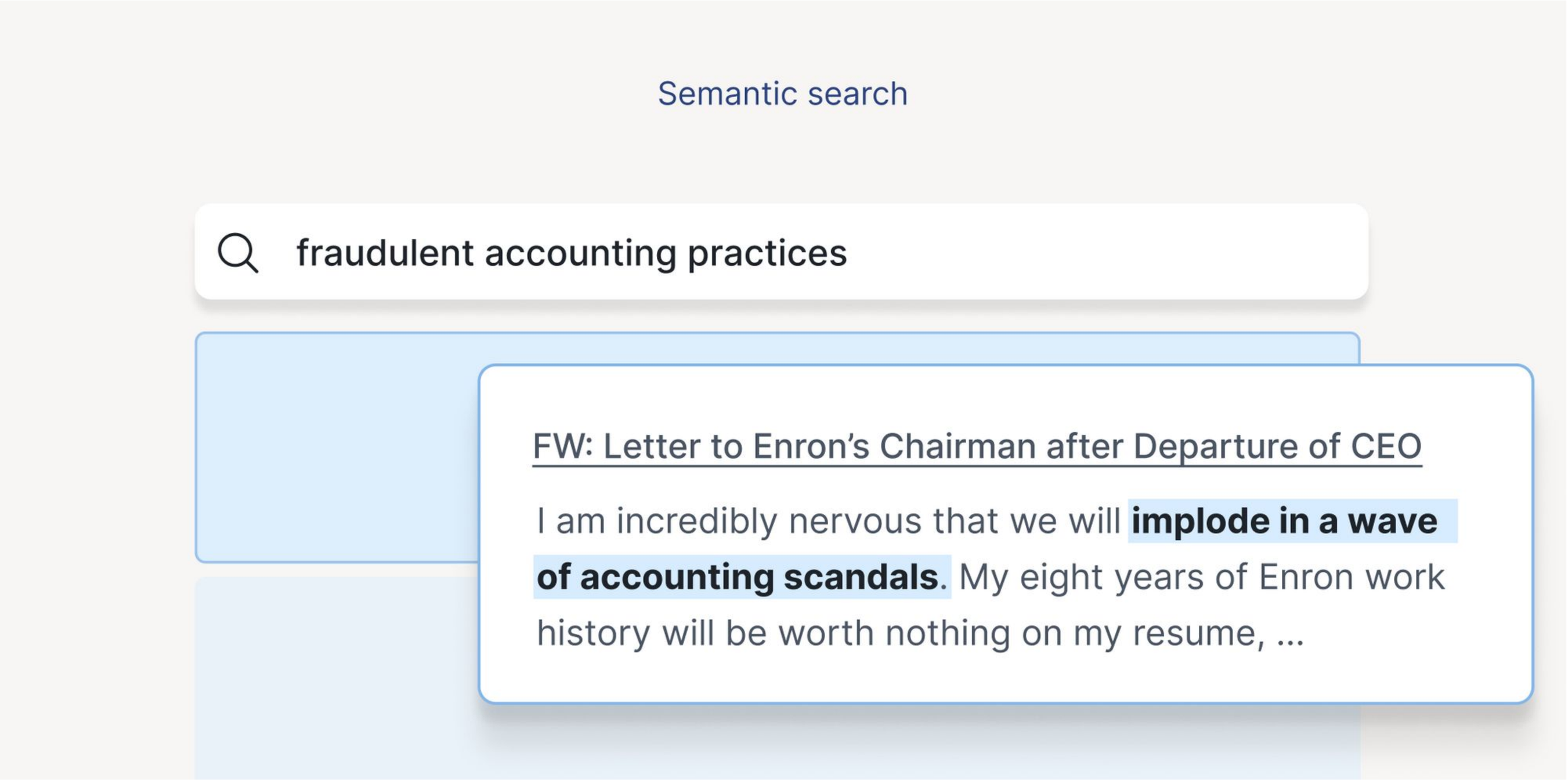


Image: DeepJudge

Modern search and retrieval utilizes semantic and lexical search

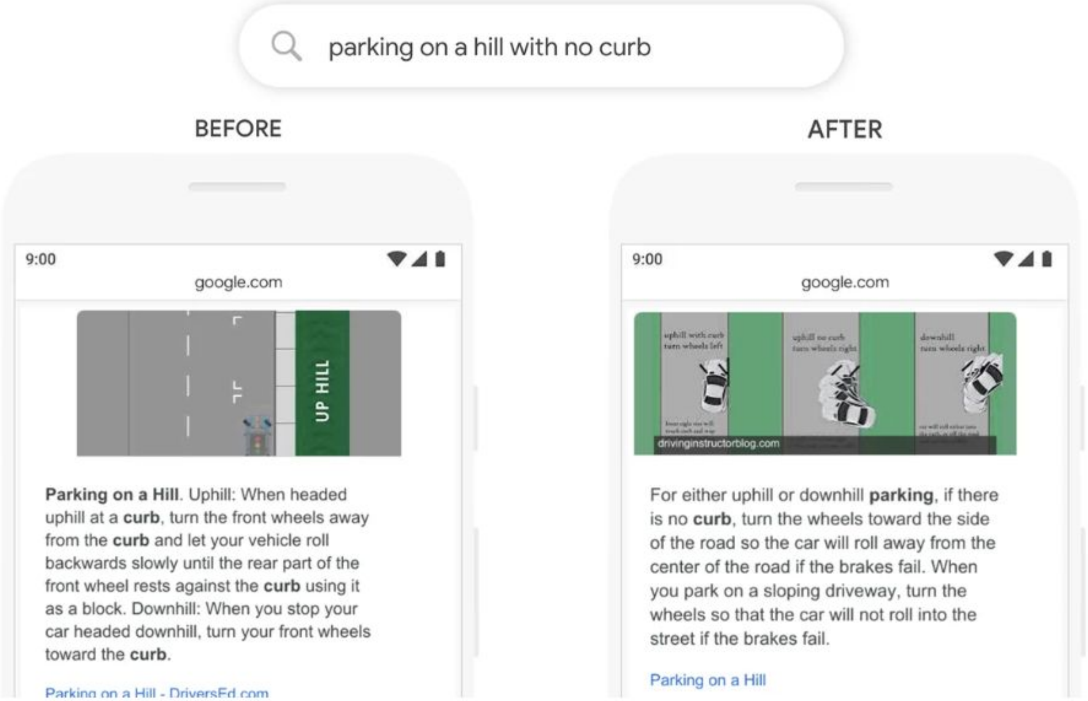
Lexical/keyword search: BM25	Semantic search: Sparse vector, dense vector	Hybrid search: RRF etc.

Semantic search: Neural/ML-learned vectors capture meaning

- Learns semantics, not *just* keyword based
- Can be fine-tuned to specific tasks and domains
- Use case scalability

For more: see my Data Day Texas 2024 talk (link)

You’ve been interacting with ML-powered semantic search for years



Powered by foundational language models e.g. BERT since [2019](#)

Image: Google

You’ve been interacting with ML-powered semantic search for years

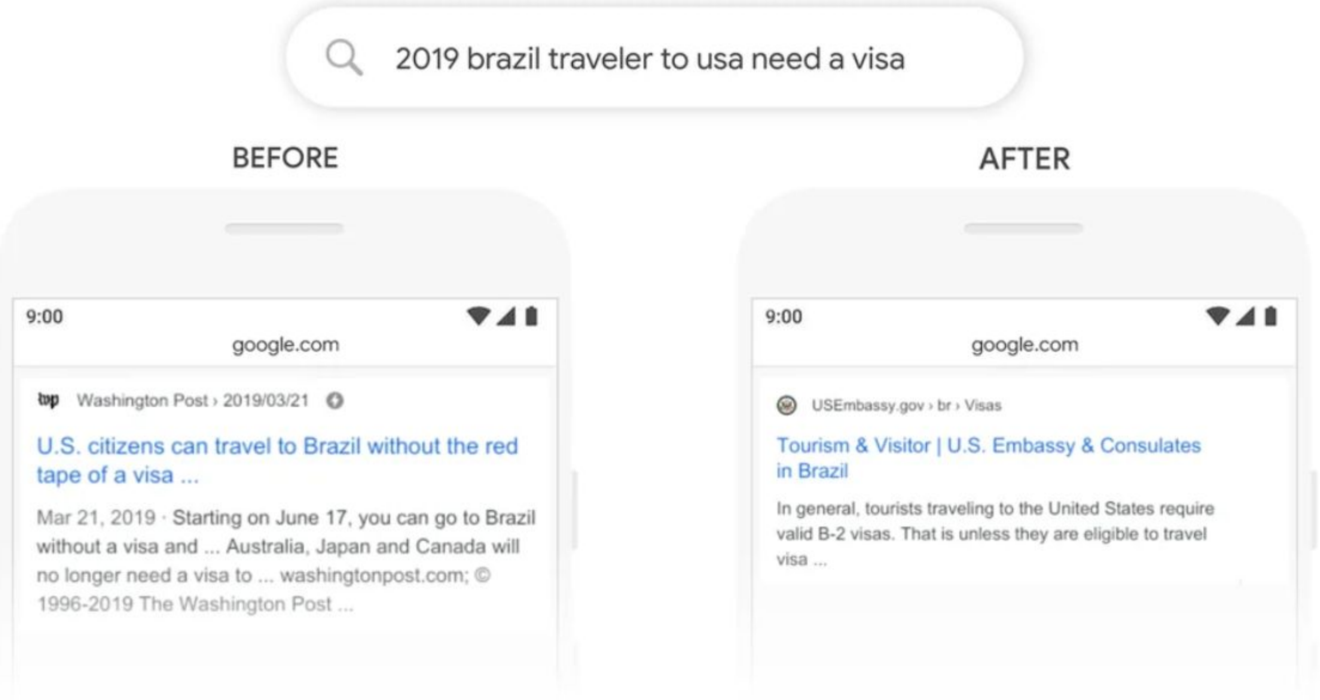
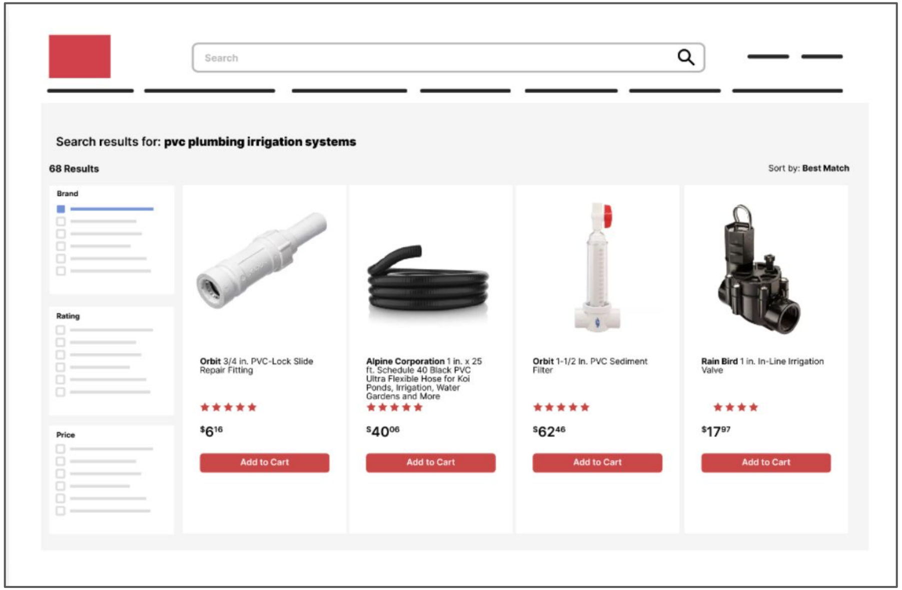


Image: Google



Mission: go from *this*

To *this*

Example: Build *new* chat function on product site with semantic search

(In addition to existing search bar)

3

Improving Retrieval: The “R” in RAG

RAG is relatively **low-hanging fruit** to provide GenAI with accurate info

Retrieve the **most relevant** [support articles / product pages / discount policies] etc…

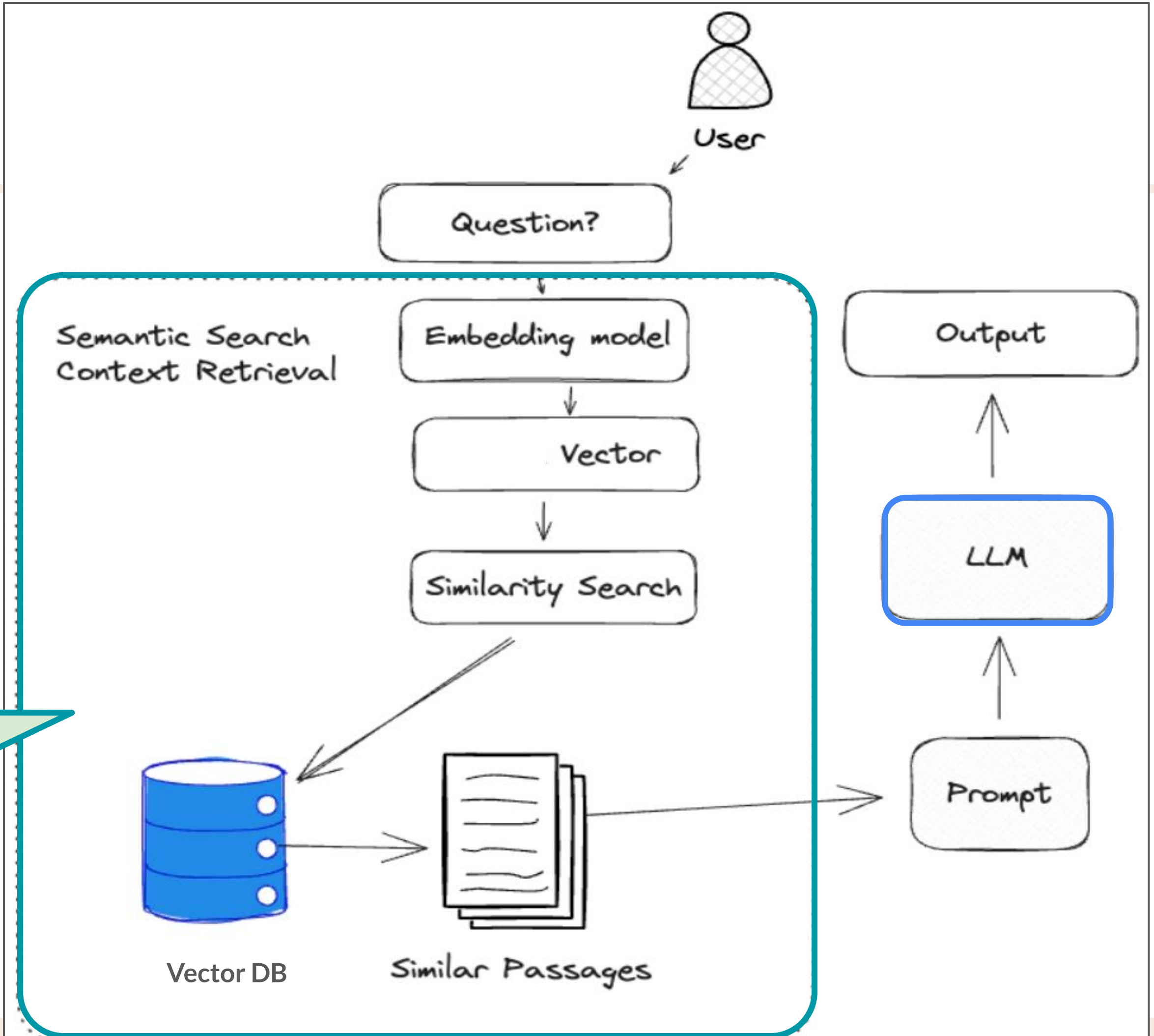
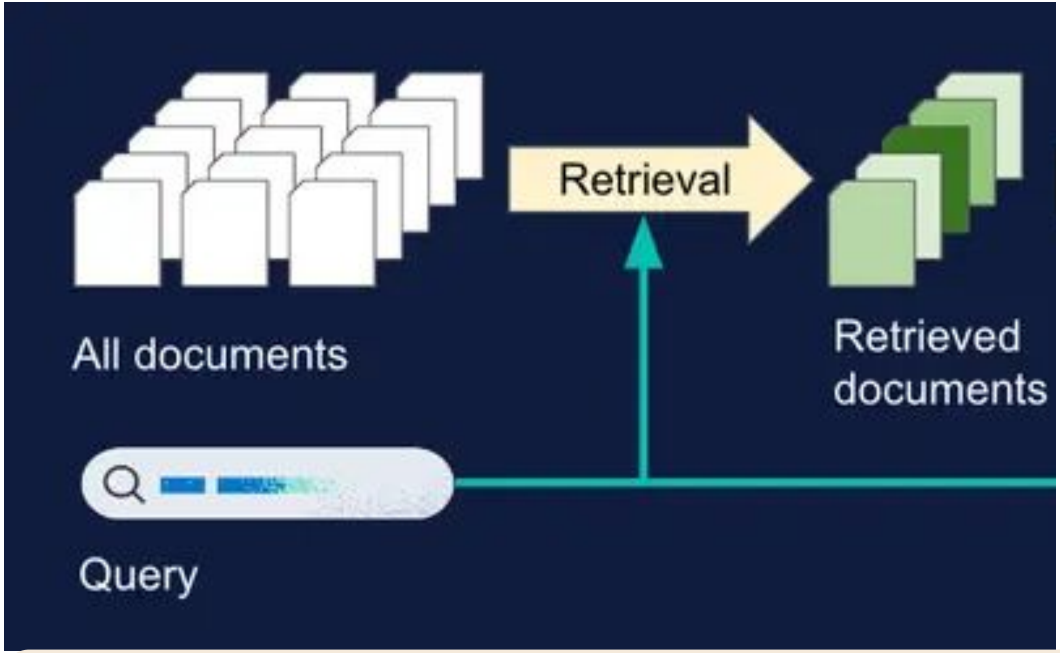


Image: Elastic

“I’ve implemented RAG, why is my LLM still outputting irrelevant answers?”

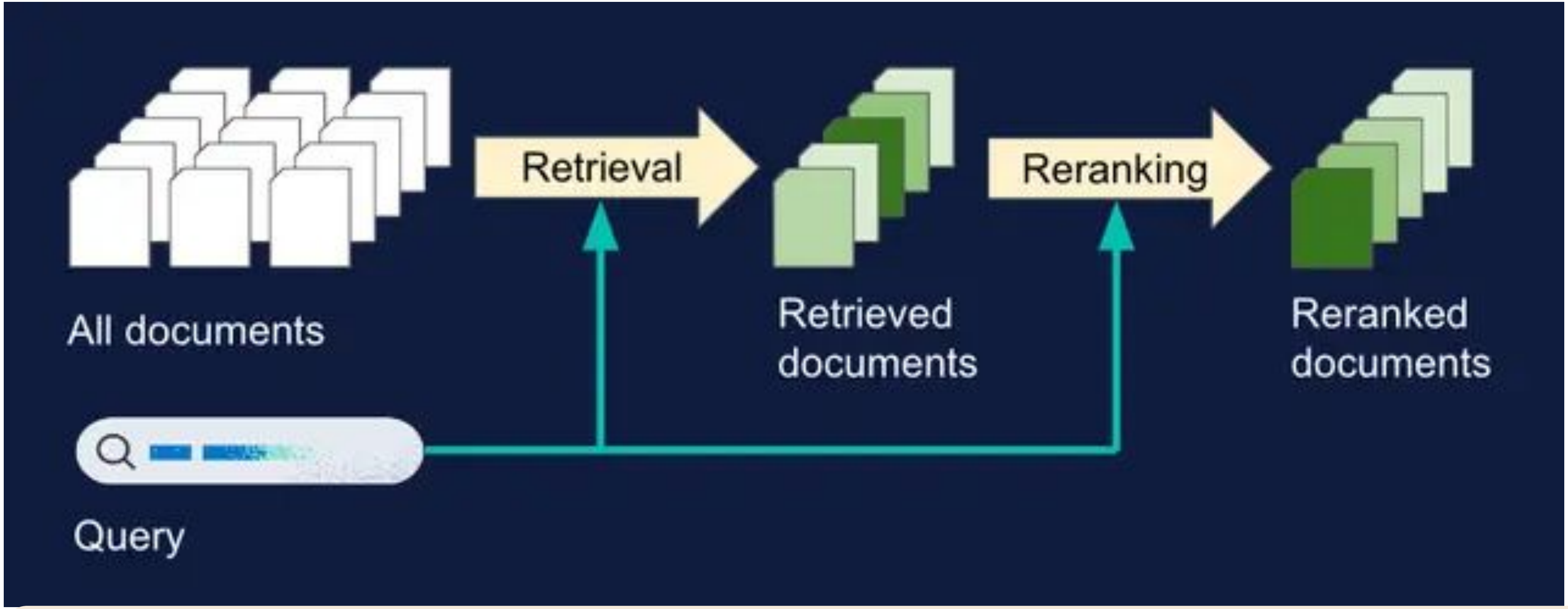
But, **retrieving** initial results is only the first step



IN PRODUCTION

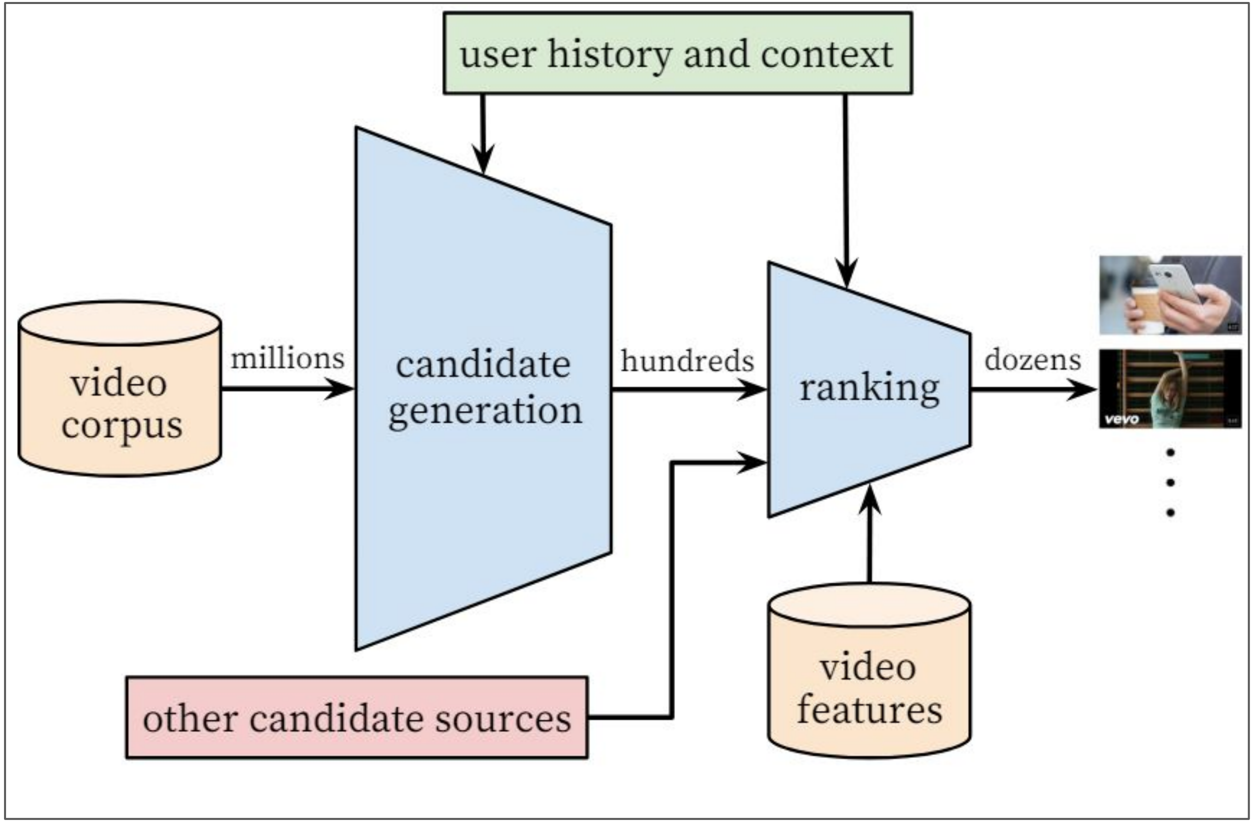
Just because some items are retrieved, doesn't mean they're displayed in the optimal order for the LLM / user

But, **retrieving** initial results is only the first step



Rerank retrieved "candidates" by importance
to pass the LLM or display to user

You’ve been interacting with reranking for a long time...



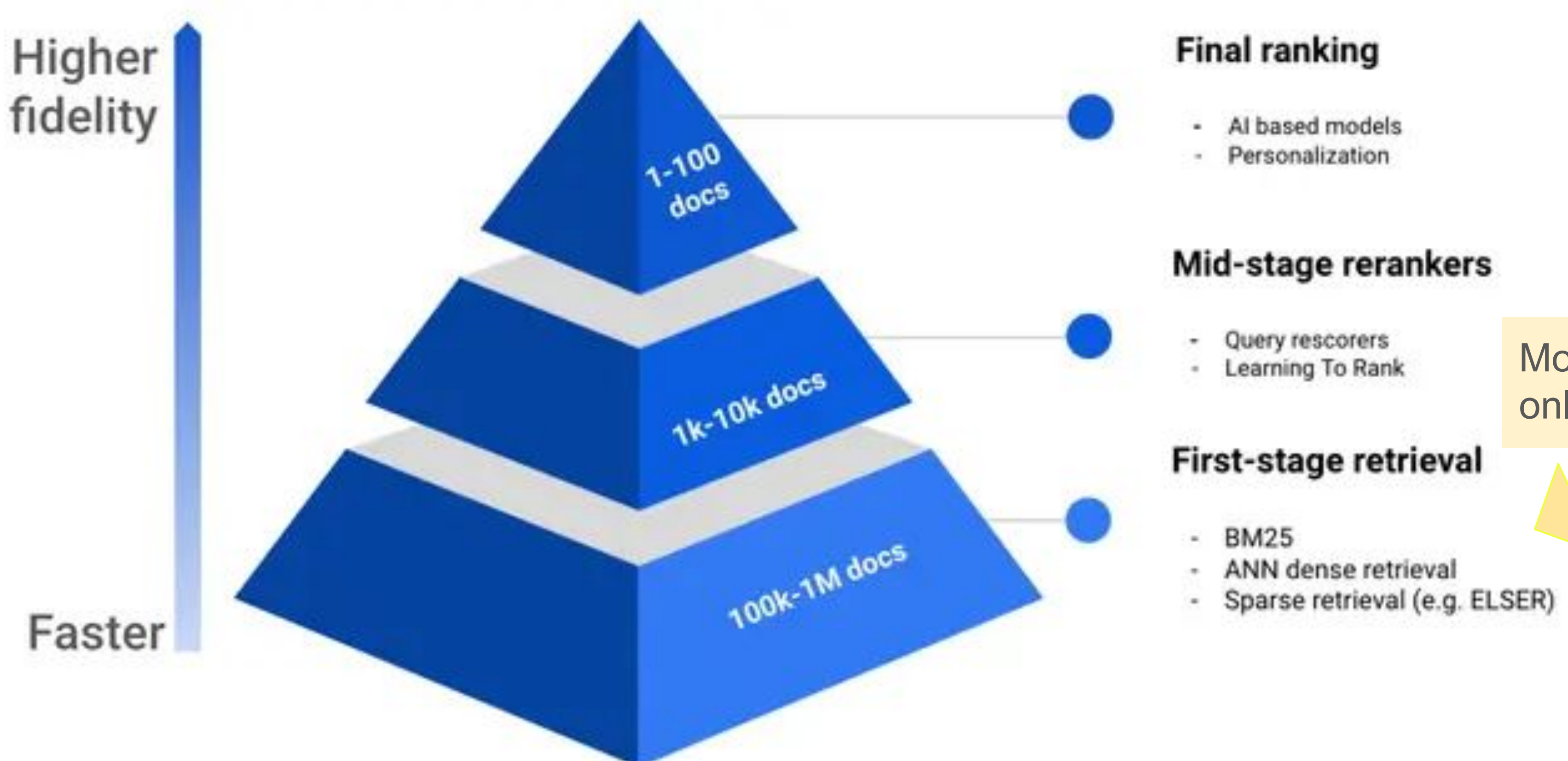
Covington, P., Adams, J.K., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. Proceedings of the 10th ACM Conference on Recommender Systems

In production, *many* considerations for **retrieval** – and RAG

- **Scale**: millions of items (videos, products, social feed…)
- **Speed**: Search over said scale of items within milliseconds
- **Fidelity:** ML models that learned more specific tasks can be more accurate

"Reranker" / multi-stage ranking is a versatile tool to address tradeoffs

Scale <> Speed <> Fidelity tradeoffs



Most doing only this

4

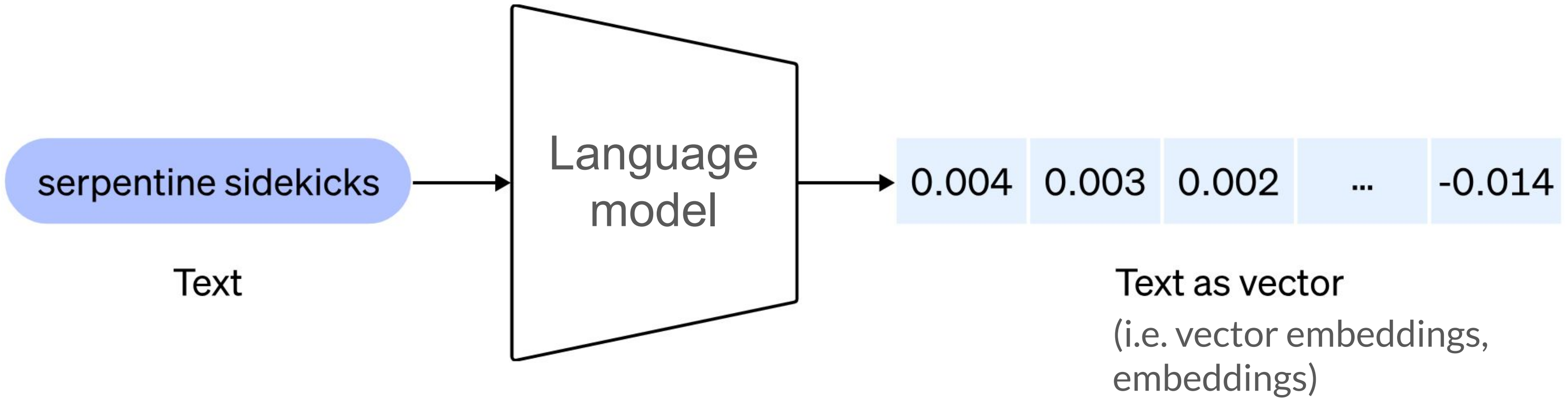
Secret sauce?! How rerankers are trained

Rerankers...

- Cohere-v3
- Elastic Rerank
- bge-reranker-v2-gemma (Google Gemma)
- mxbai-rerank-base-v1
- monot5-large
- MiniLM-L12-v2

etc.

ML: Text is now turned into vectors / represented by *numbers*



Based on
Image: OpenAI

Machine learning models are happy. Love working with numbers.

Dataset

QA pairs

Negative mining

Training

Cross-encoder

Model outputs

Building blocks for many rerankers, with examples from Elastic Rerank training

Datasets

- BEIR – Benchmarking IR
- MTEB – Massive Text Embedding Benchmark
- MS MARCO

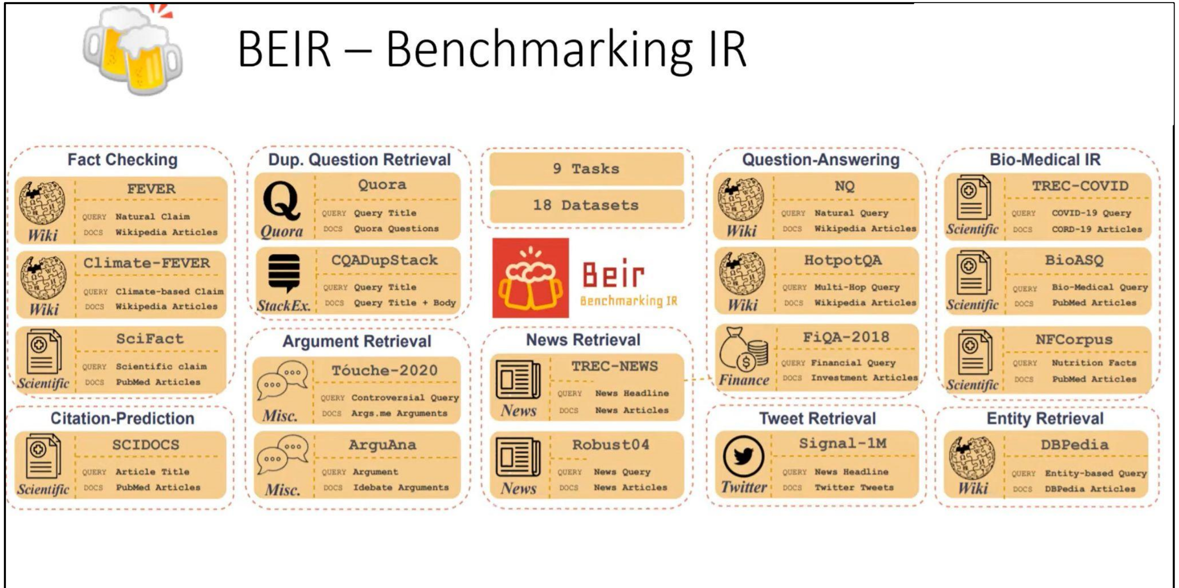


Image: BEIR

MS MARCO: common question-answering dataset

MS MARCO

Microsoft

Additional datasets for tasks such as passage ranking, keyphrase extraction, language generation etc. have been subsequently added

MS MARCO dataset contains Bing questions + human generated answer

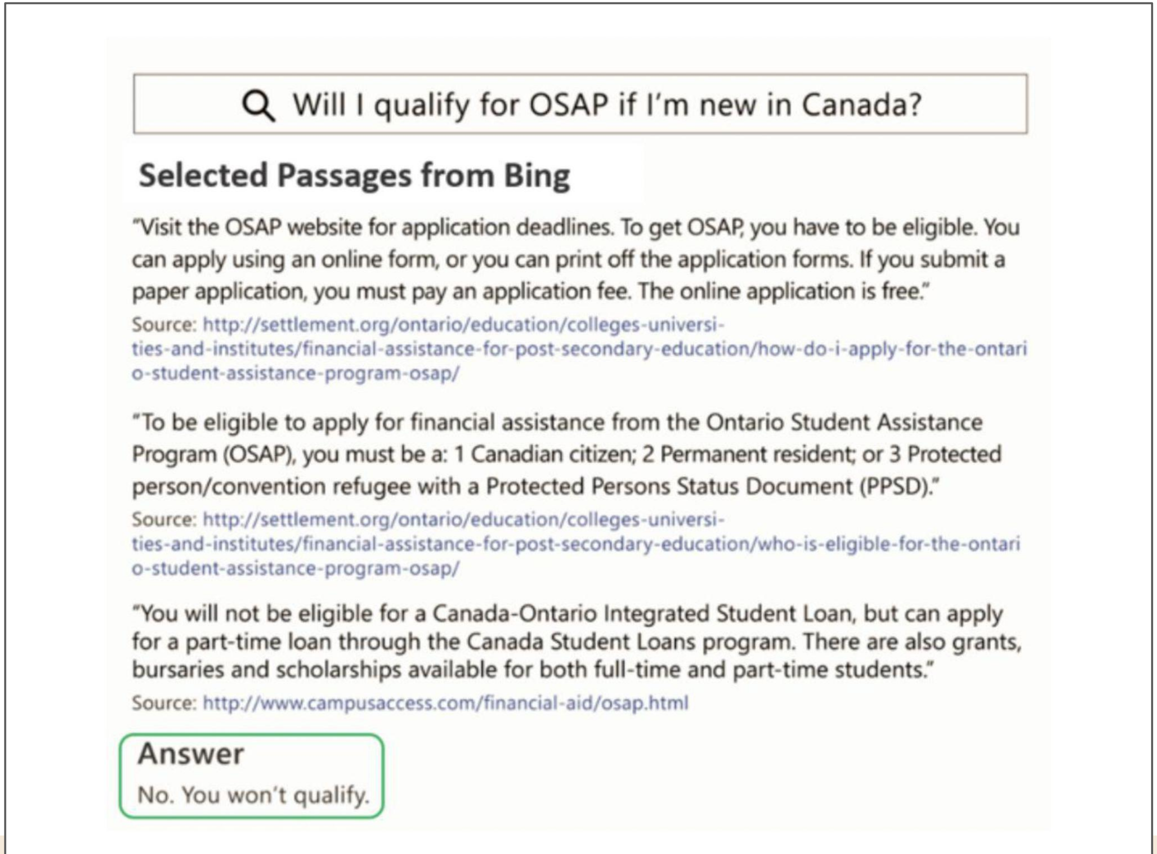


Image: MS MARCO

Dataset generation (Elastic Rerank)

- Total: 3 million queries
- Open QA datasets + 180,000 synthetic pairs
- Diverse query types
  - Keyword search
  - Exact phrase matching
  - Short and long natural language questions

Present model with more diverse scenarios:
“Deep-negative mining with multiple negatives per query”

Contrastive learning

- ❏ Bring together representations of similar examples
- ❏ Push apart representations of differing examples

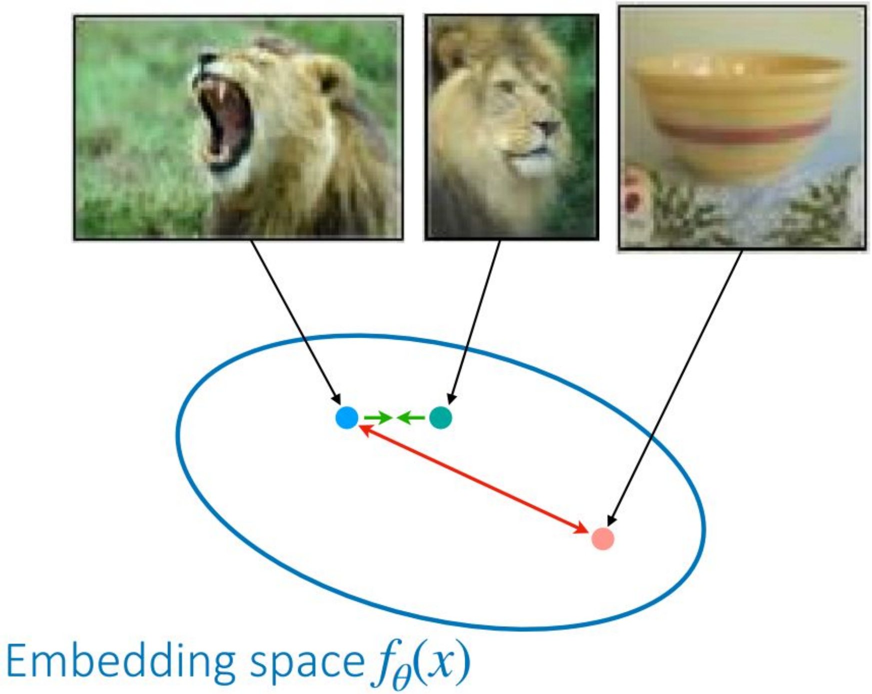


Image: Stanford

Bi-encoder and cross-encoder: different ways to structure encoders

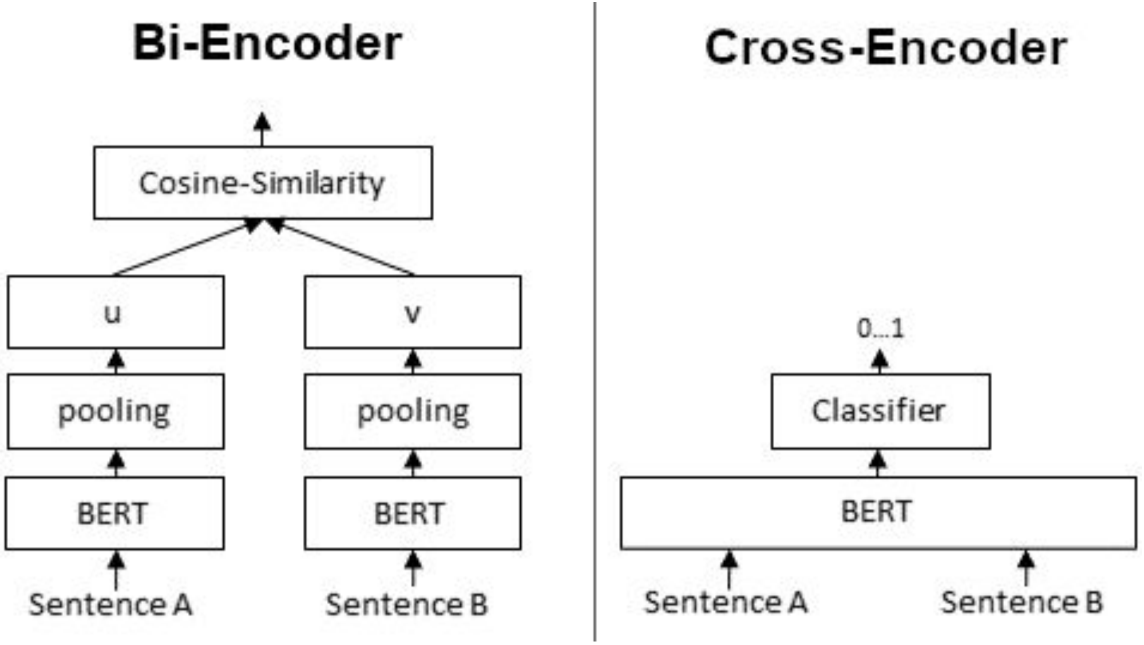
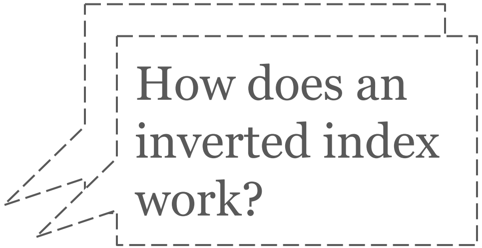
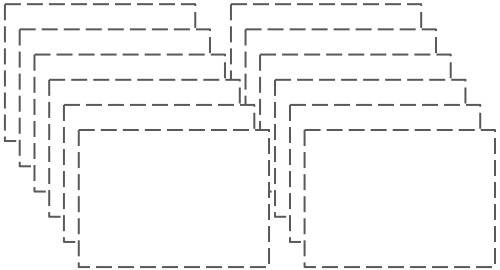


Image: Sentence-Transformers

Bi-encoders allow for query and answer to be encoded separately



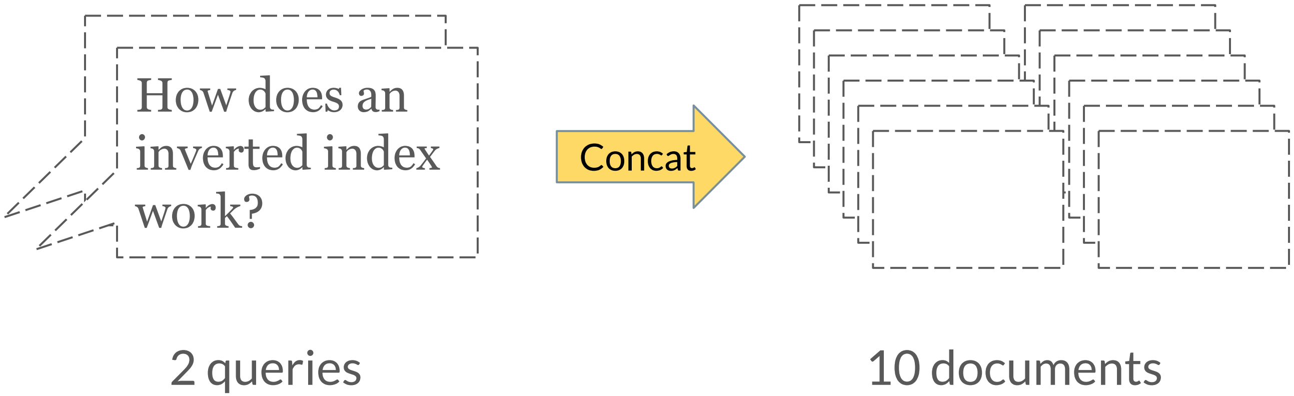
2 queries



10 documents

Bi-encoder: encode 12 times; can pre-encode 10 documents

Cross-encoders are more accurate, but slower



Cross-encoder: encode 20 times on query time

Cross-encoders vs. Bi-encoders

Cross-encoders

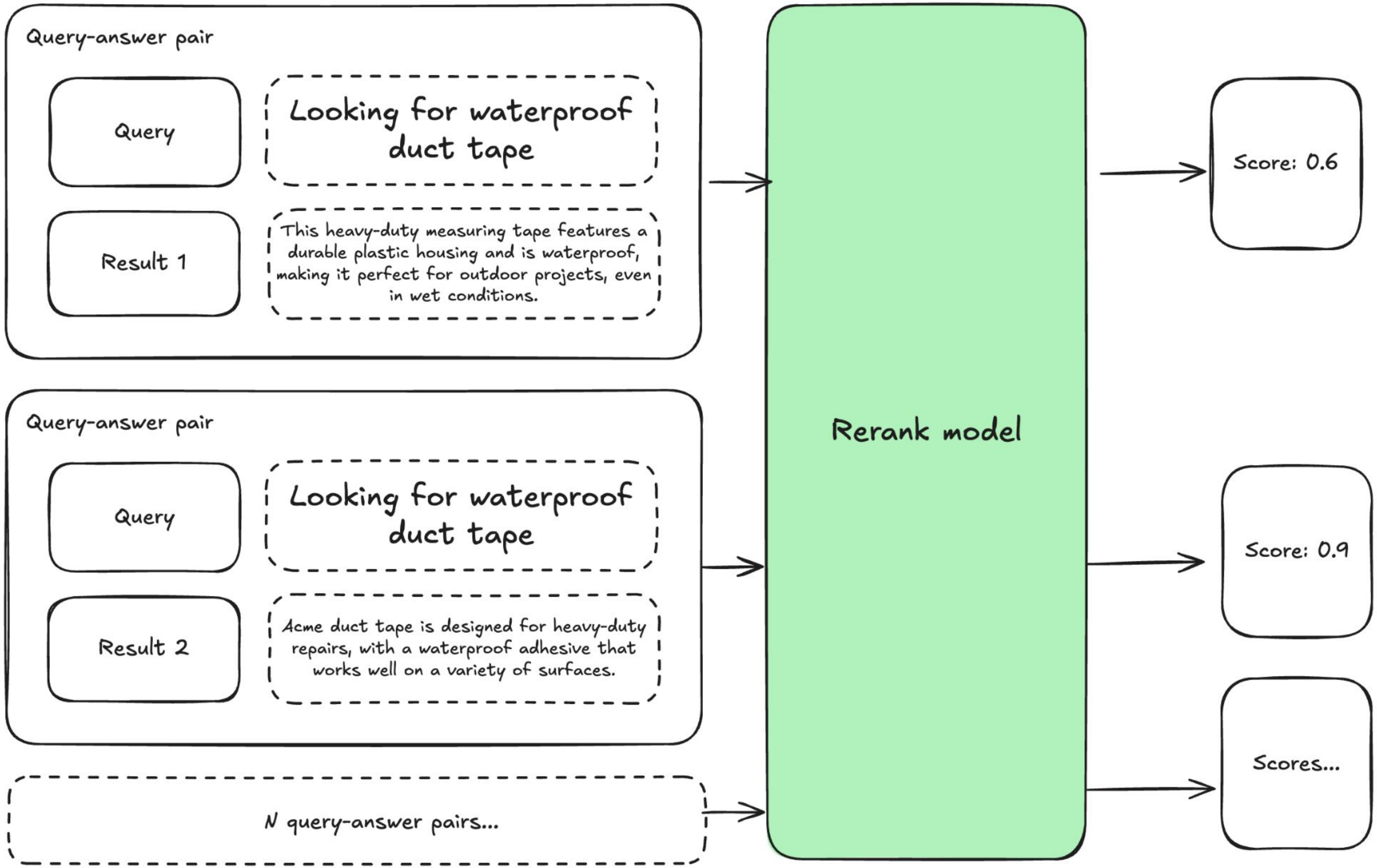
- Can learn more robust representations for generally assessing relevance
- Captures more nuanced semantics: Cross-encoder models can better learn how negation should affect relevance judgments
- Better calibrated across a diverse range of query types and topics. This makes choosing a score at which to drop documents significantly more reliable.

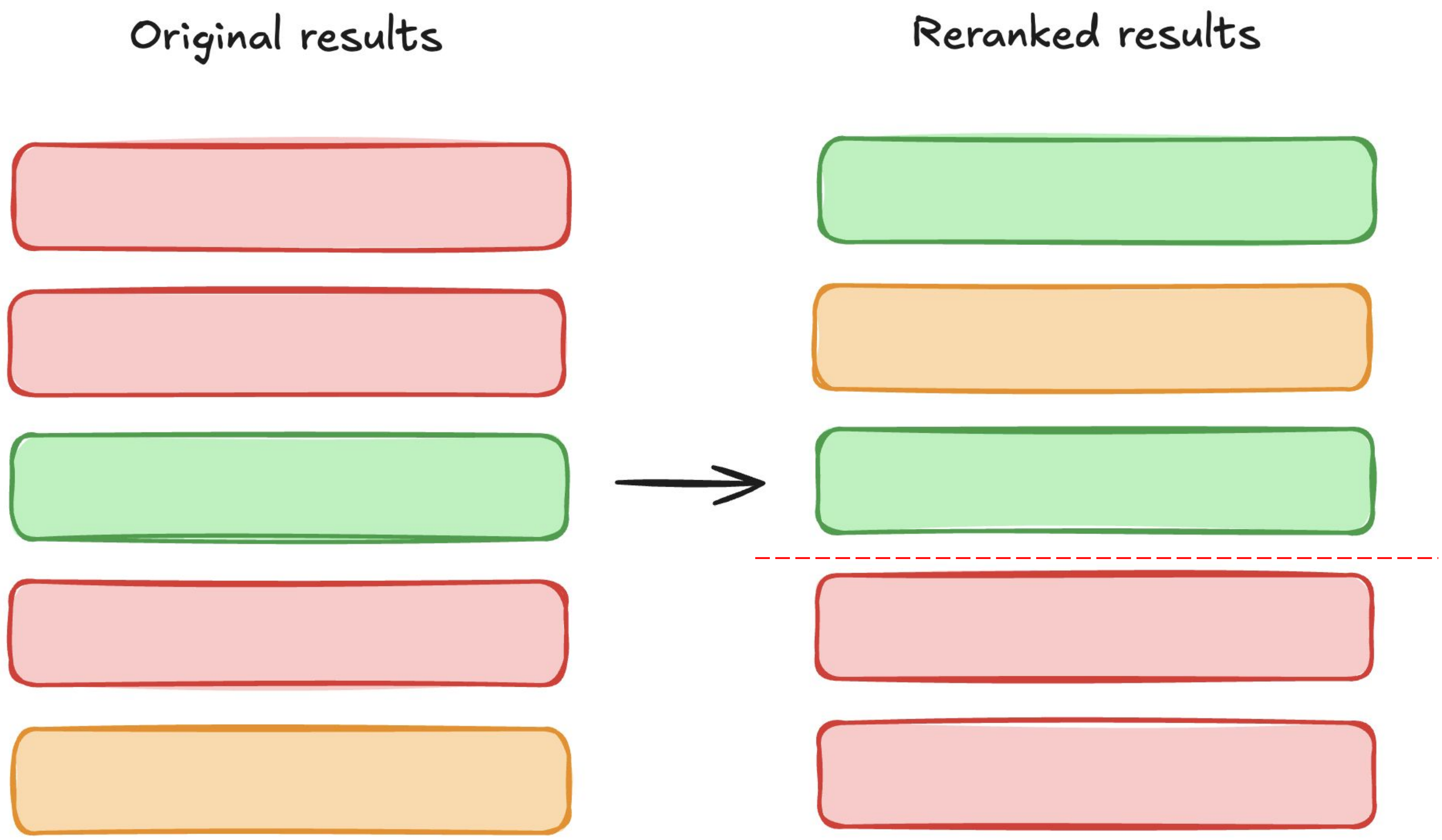
Bi-encoders

- Bi-encoder models struggle more with things like negation and instead tend to pick up on matches for the majority concepts in the text, independent of whether the query wants to include or exclude them.
- Less encoding needed

STANLEY
FATMAX
8m
33-728

DUCT TAPE


Query-answer pair
Query
Looking for waterproof duct tape
Result 1
This heavy-duty measuring tape features a durable plastic housing and is waterproof, making it perfect for outdoor projects, even in wet conditions.
Query-answer pair
Query
Looking for waterproof duct tape
Result 2
Acme duct tape is designed for heavy-duty repairs, with a waterproof adhesive that works well on a variety of surfaces.
N query-answer pairs...
Rerank model
Score: 0.6
Score: 0.9
Scores...


Original results
Reranked results

Example

{

"query": "What duct tape is waterproof?",

"input": [

"This heavy-duty **measuring tape** features a durable plastic housing and is **waterproof**, making it perfect for outdoor projects, even in wet conditions.",

"Our compact **measuring tape** is built with a weather-resistant coating, but it is **not** fully **waterproof**. It's ideal for general indoor and light outdoor use.",

"Acme **duct tape** is designed for heavy-duty repairs, with a **waterproof** adhesive that works well on a variety of surfaces.",

"This **tape** is ideal for office use, offering a strong adhesive and a clear finish. While it's *resistant to moisture*, it is not suitable for outdoor use or measuring purposes.",

"This **waterproof measuring tape** is engineered for construction and outdoor use, featuring a non-slip rubberized casing and a clear, easy-to-read scale that resists moisture and rust."

]

}

Model outputs: scores for each document wrt query


```
{
 "rerank": [
  {
   "index": "3",
   "relevance_score": "0.99838966"
  },
  {
   "index": "1",
   "relevance_score": "0.587174"
  },
  {
   "index": "0",
   "relevance_score": "0.061199225"
  },
  {
   "index": "2",
   "relevance_score": "0.032283258"
  },
  {
   "index": "4",
   "relevance_score": "0.015365342"
```


IN PRODUCTION

Use relevance scores **cutoff** to **prune** irrelevant documents.

Use scores for **observability**: window into how well your RAG is performing.

- "Why is my average relevancy score for these types of queries so low?". By contrast, without these metrics you're in the dark when it comes to optimizing your RAG pipeline.

5

All together: Rerankers in RAG systems

Retrieve the **most relevant** [support articles / product pages / discount policies] etc…

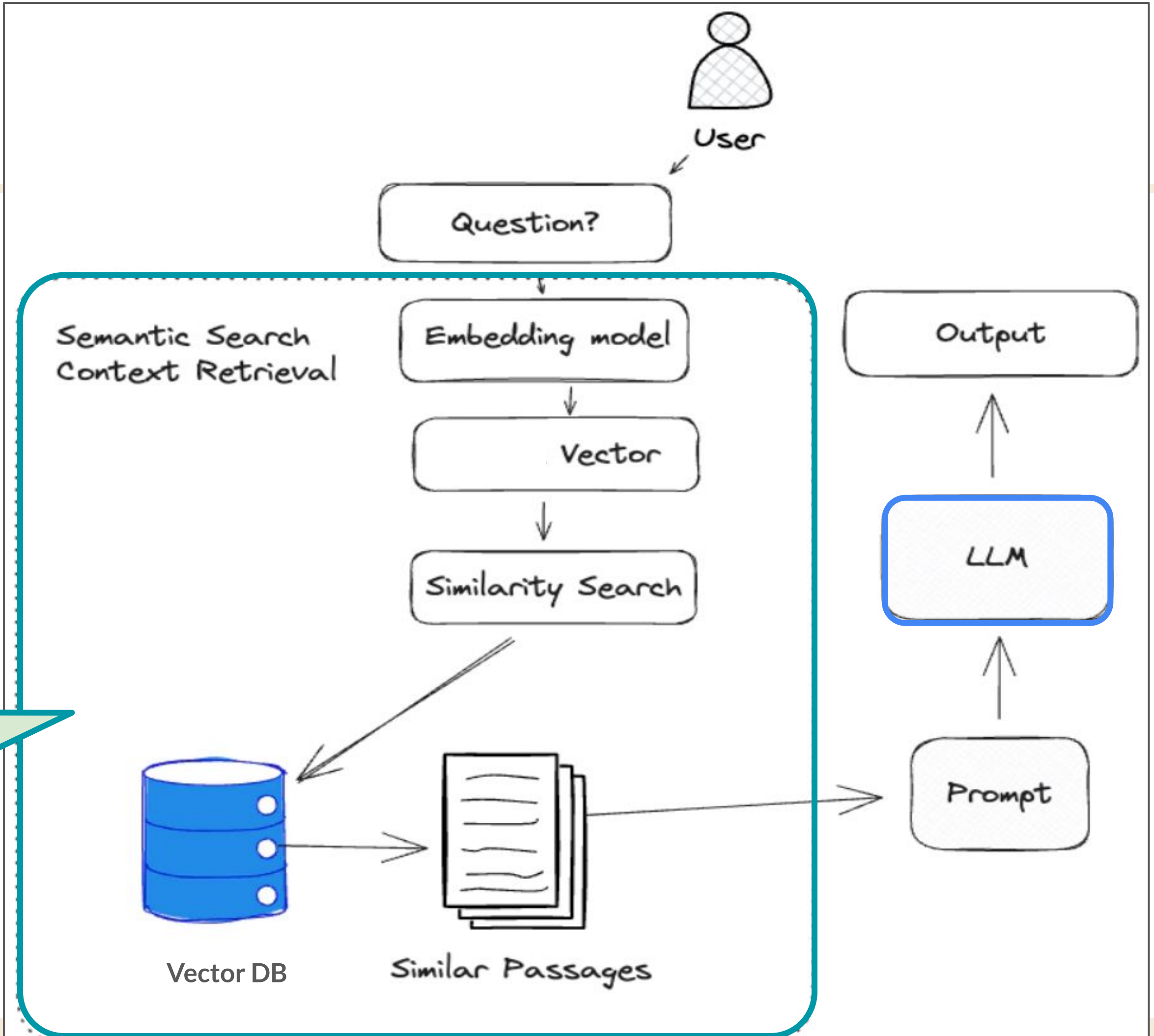
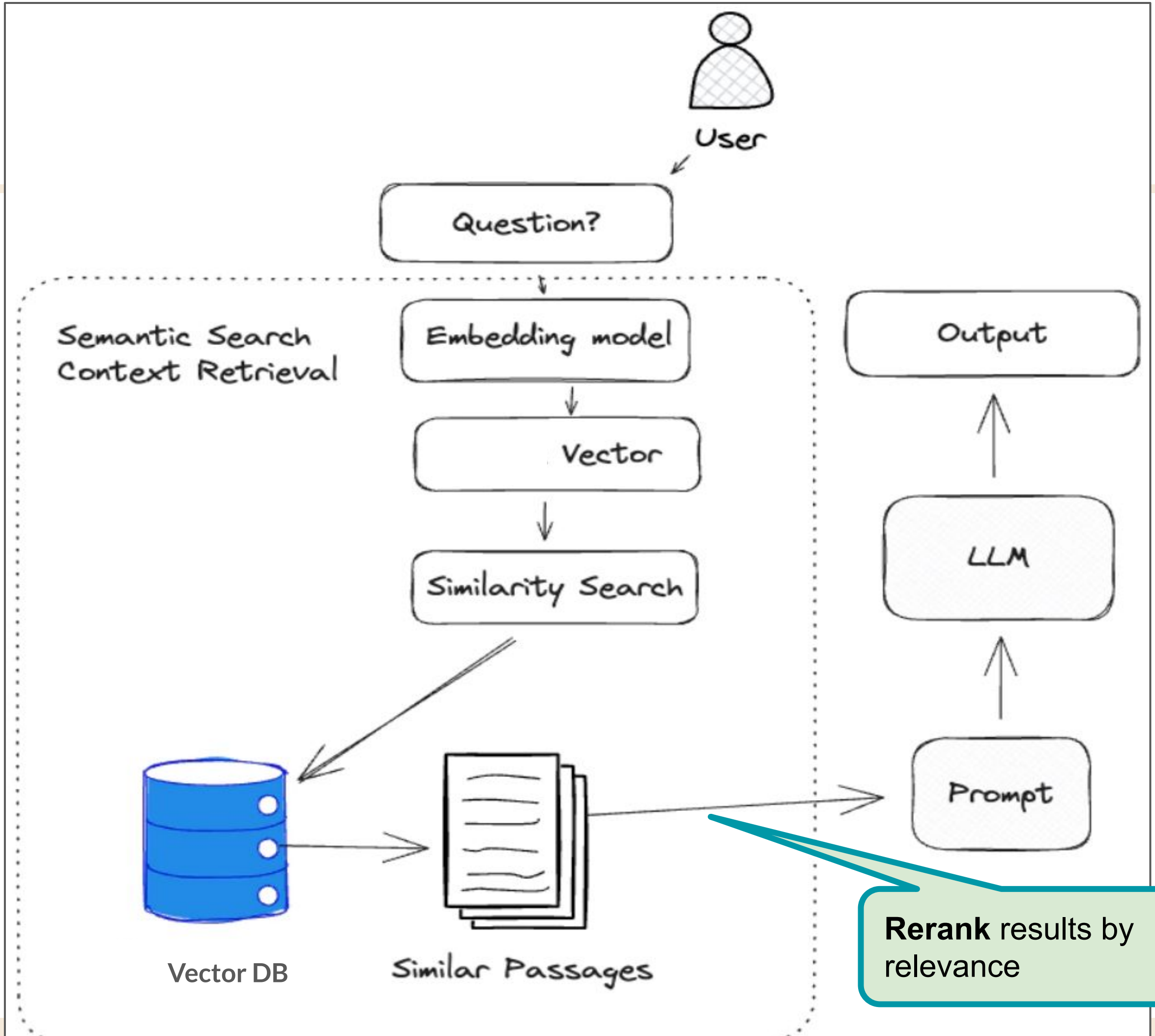
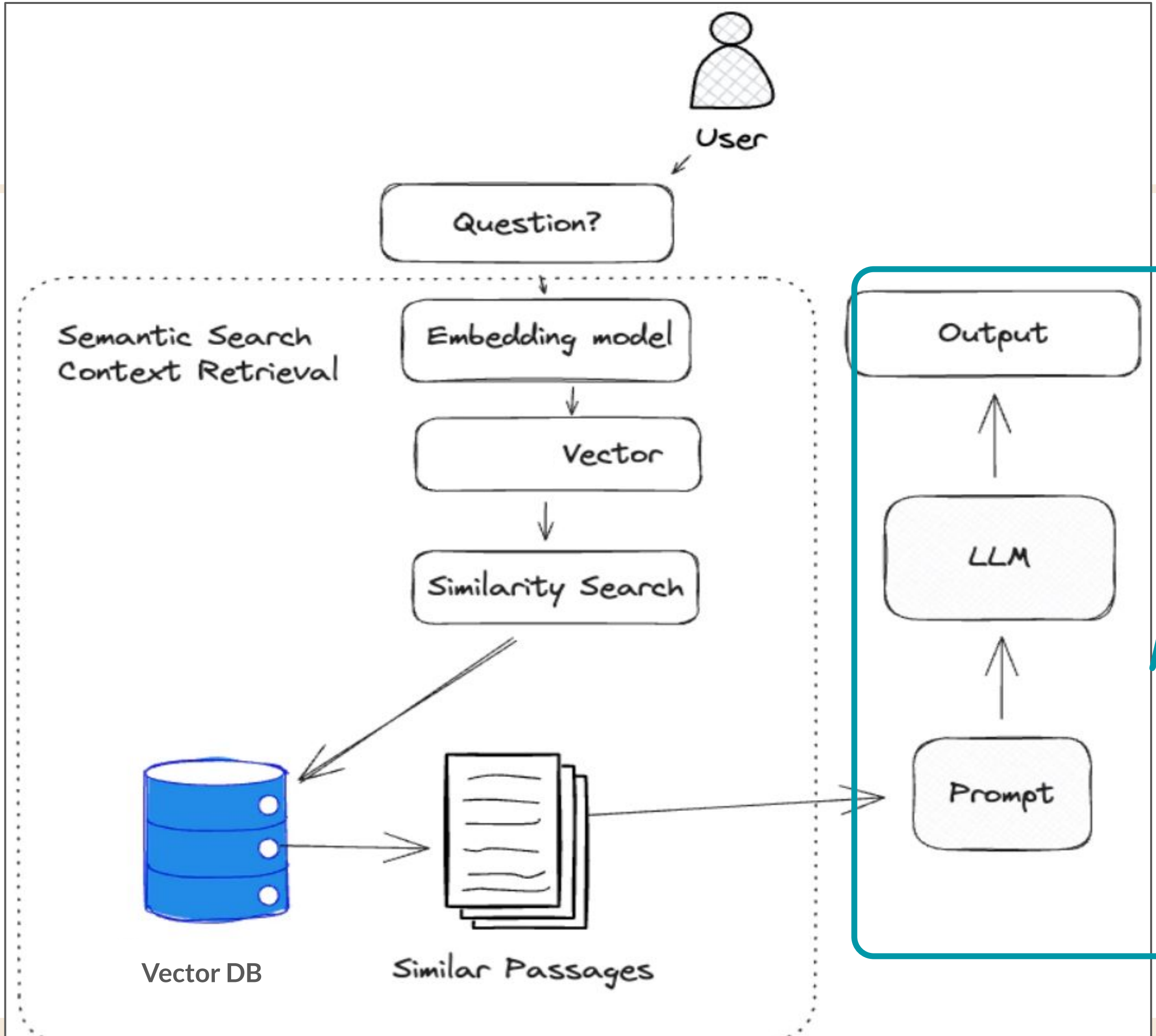


Image: Elastic



Rerank results by relevance

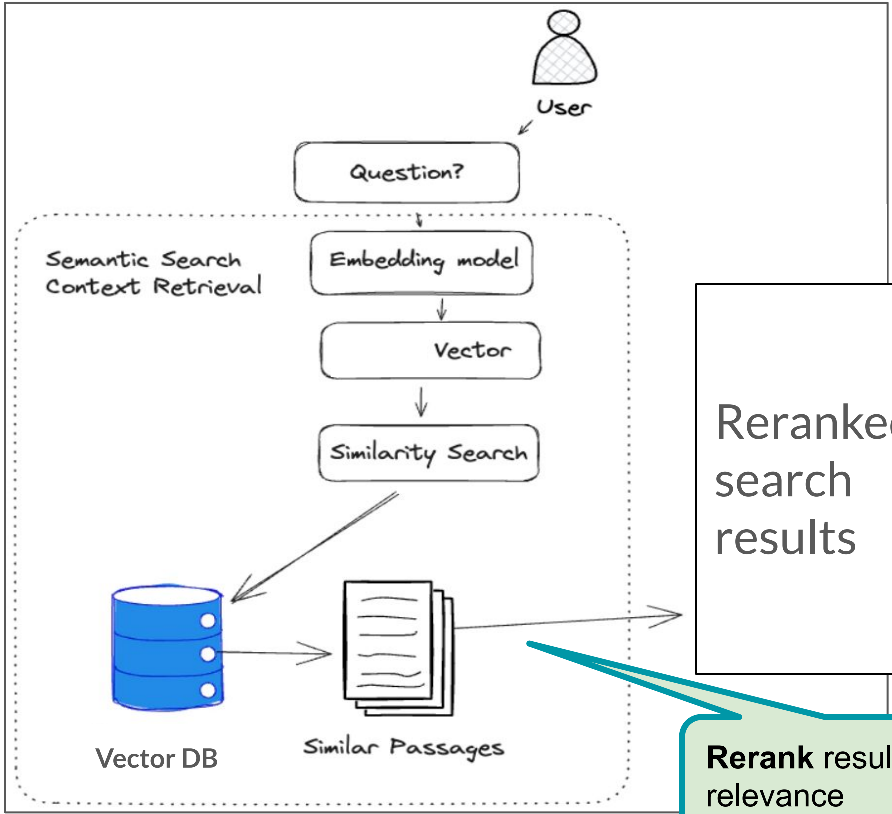
Image: Elastic



Pass reranked info to LLM/ Chatbot to **generate** the final output

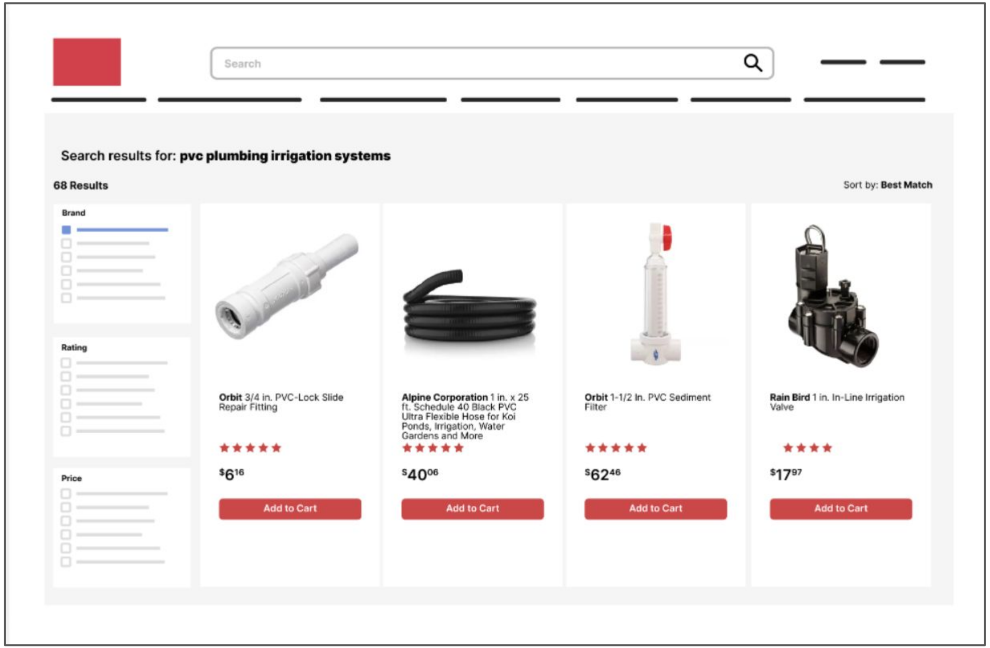
Image: Elastic

Reranking: Helps your plain ol' search results (if you're not using RAG)



Reranked search results

Rerank results by relevance



BM25 is still a strong baseline; tuning to compete for user experience

Conclusion & Going forward

1. Use reranking to improve results (the “R” in RAG)

Conclusion & Going forward

1. Use reranking to improve results (the “R” in RAG)
2. Speed <> Fidelity <> Scale tradeoffs
    - Faster model to filter, Slower but more fine-grained model to do final re-ranking

Conclusion & Going forward

1. Use reranking to improve results (the “R” in RAG)
2. Speed < > Fidelity < > Scale tradeoffs
    - Faster model to filter, Slower but more fine-grained model to do final re-ranking
3. Tune and improve individual GenAI system components
    - Retrieval: **Keyword** or **semantic** or **hybrid** e.g. Reciprocal Rank Fusion (RRF)
    - Semantic retrieval: **Sparse** (e.g. ELSER) or **dense** (e.g. E5)
    - Reranker, fine tuning, etc.

Conclusion & Going forward

1. Use reranking to improve results (the “R” in RAG)
2. Speed < > Fidelity < > Scale tradeoffs
   - Faster model to filter, Slower but more fine-grained model to do final re-ranking
3. Tune and improve individual GenAI system components
   - Retrieval: **Keyword** or **semantic** or **hybrid** e.g. Reciprocal Rank Fusion (RRF)
   - Semantic retrieval: **Sparse** (e.g. ELSER) or **dense** (e.g. E5)
   - Reranker, fine tuning, etc.

Q&A

Find me at:

- **Elastic booth** or **O'Reilly booth**
- LinkedIn: Susan Shu Chang
- Read more about our work: Elastic blog

Get the **Machine Learning Interviews** book or recommend to someone you know:

https://amzn.to/4aOjO26

(Limited copies available at DDT O'Reilly booth)