Search Medium Write

Hybrid search with Re-ranking

Sowmiya Jaganathan · Following
4 min read · Jul 24

71 2

When it comes to setting up search algorithms, we rely on multiple techniques to ensure we get the most relevant results. However, combining the results of different methods can be a challenge.

Let's take two major approaches for our experiment: semantic similarity and statistical methods like BM25 or TF-IDF. Semantic similarity and statistical methods have their strengths in retrieving information, depending on the query type. But finding a balance to prioritize which results should be displayed under specific circumstances isn't easy.

In this scenario, directly sorting the search results becomes tricky due to the mismatch in document score ranges. Keyword search scores fall within a positive range, usually between 0 and some maximum value, depending on the relevance of the query. On the other hand, semantic searches (*eg,.with cosine similarity*) generate document scores between 0 and 1.

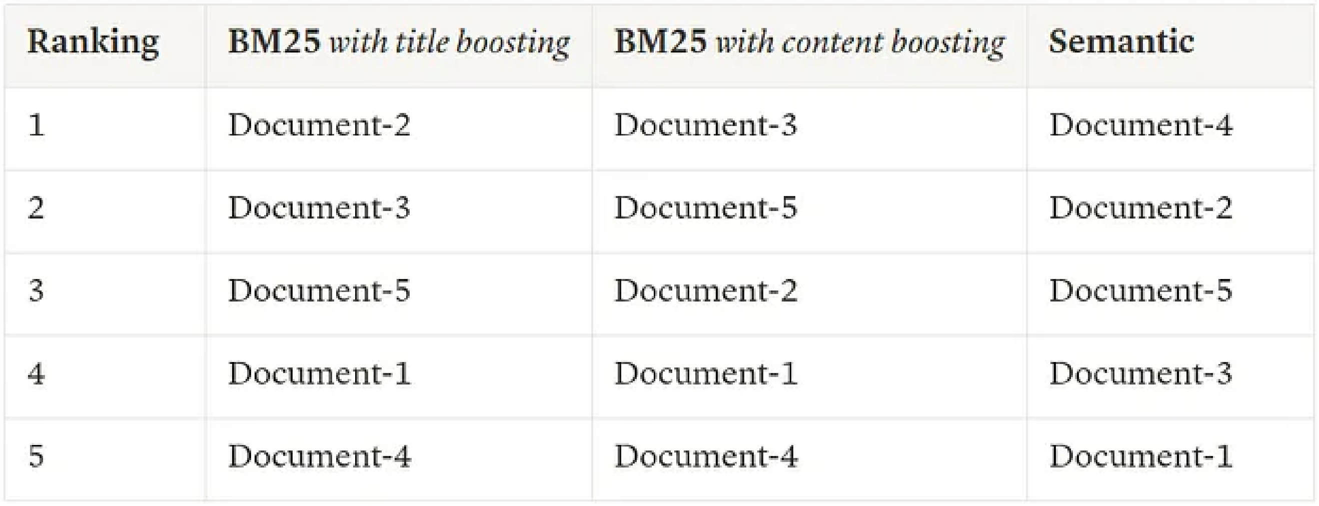
So, Today, we'll explore how to effectively combine the results with re-ranking techniques.

Reciprocal Rank Fusion (RRF)

Reciprocal Rank Fusion (RRF) is a method used to combine results from different retrieval systems by leveraging the positions/rank of the documents.

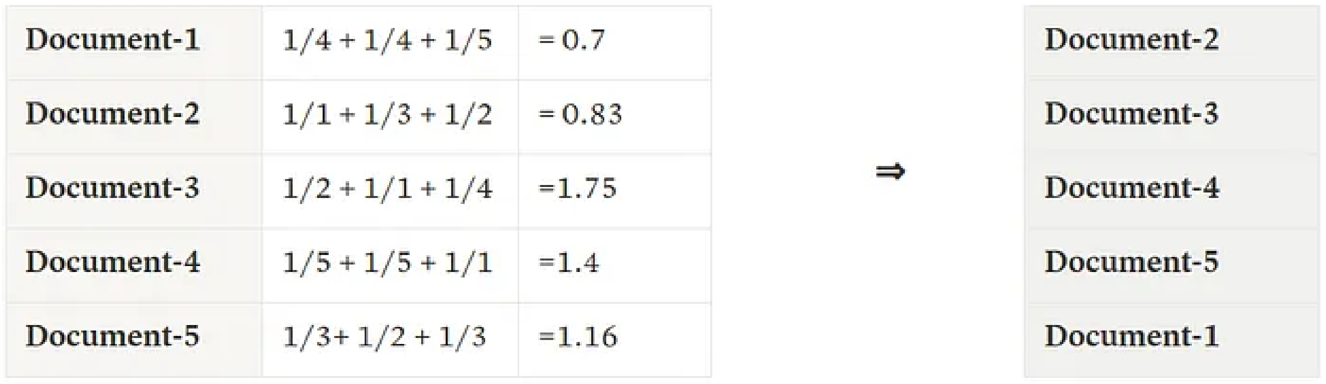
```
RRFscore(d ∈ D) = Σ [1 / (k + r(d))]

# k is a constant that helps to balance between high and low ranking.
# r(d)is the rank/position of the document
```

Let’s understand with an example.



Ranking	BM25 *with title boosting*	BM25 *with content boosting*	Semantic
1	Document-2	Document-3	Document-4
2	Document-3	Document-5	Document-2
3	Document-5	Document-2	Document-5
4	Document-1	Document-1	Document-3
5	Document-4	Document-4	Document-1

Let’s calculate RRF for each document and rerank:



Document-1	1/4 + 1/4 + 1/5	= 0.7
Document-2	1/1 + 1/3 + 1/2	= 0.83
Document-3	1/2 + 1/1 + 1/4	=1.75
Document-4	1/5 + 1/5 + 1/1	=1.4
Document-5	1/3+ 1/2 + 1/3	=1.16

⇒

Document-2
Document-3
Document-4
Document-5
Document-1

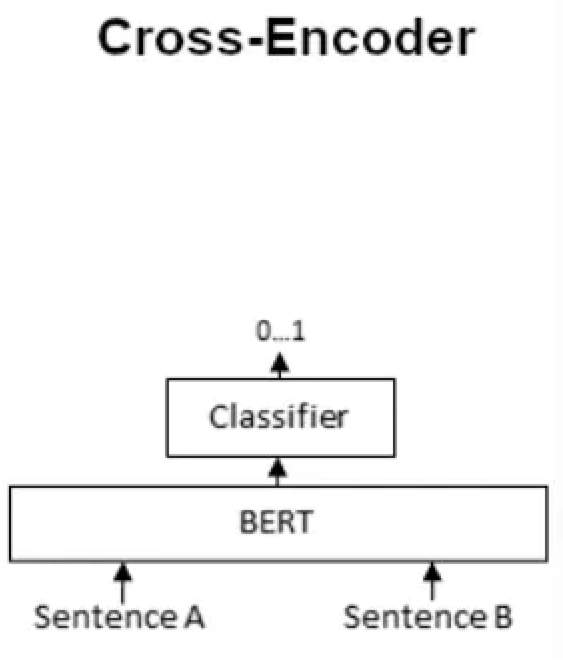
In the reranked table, we could see Document-2 is in the top position followed by document-3 and document-4.

Re-ranking with Cross-Encoder model

Let's recap Bi-Encoder models: The documents are encoded and stored in the embedding space. When the user query comes in, query is encoded at the run time to calculate similarity distance between the documents and the query.

> *Now, with the Cross-Encoder model, we pass the query and document simultaneously. The model encodes and generates contextual embeddings for each word in the input sequences.* ***It combines the contextual embeddings of both sequences to create a joint representation that captures the interactions between the input pairs.*** *Then, the classification layer uses this joint representation to predict whether they are related or not related based on the interactions.*

This leads to more accurate representations and better performance than Bi-encoder.



sbert

Let’s understand this with an examples: With the list of corpus, we re-ranked them based on the scores obtained from both the Cross Encoder and Bi Encoder models.

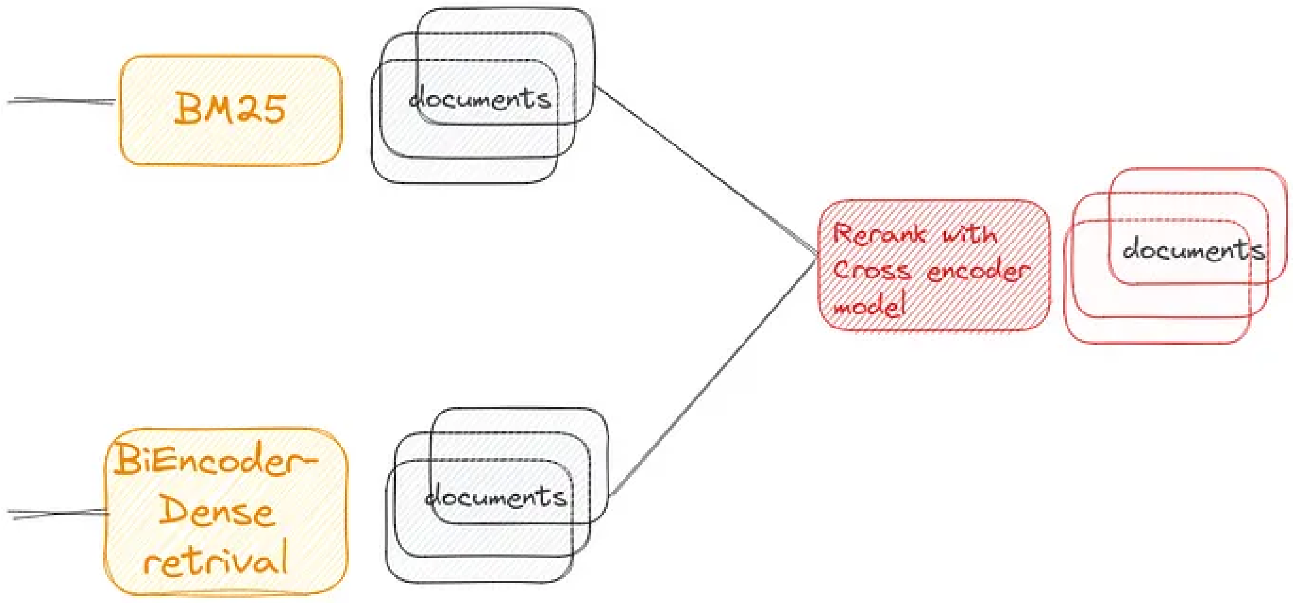
```
#sample script
from sentence_transformers.cross_encoder import CrossEncoder
model = CrossEncoder('model_name_or_path')
scores = model.predict([["text", "sentence pair"],
                        ["text", "sentence pair"]])
```

```
# Cross-Encoder
Query: king rules the country
0.75    The monarch holds sway in the nation.
0.59    Majesty commands the nation.
0.59    Sovereign reigns over the realm.
0.58    Ruler leads the land.
0.51    Emperor controls the country.
0.50    Monarch governs the nation.
0.27    The country cherishes its royal heritage.
```

```
#Bi-Encoder
Query: king rules the country
0.67    Monarch governs the nation.
0.63    The monarch holds sway in the nation.
0.52    Ruler leads the land.
0.49    Emperor controls the country.
0.47    Majesty commands the nation.
0.46    Sovereign reigns over the realm.
0.46    The country cherishes its royal heritage.
```

While both were able to perform good except the last sentence which conveyed a different meaning despite being semantically similar to query. Cross-Encoder were able to capture the difference, while the Bi Encoder provided a similar score as the sentences were semantically similar.

Now we understood the concept, let’s see the full picture once.



References:

Reciprocal rank fusion | Elasticsearch Guide [8.8] | Elastic

This functionality is in technical preview and may be changed or removed in a future release. Elastic will apply best...

www.elastic.co

Using Cross-Encoders as reranker in multistage vector search | Weaviate - vector database

Learn about bi-encoder and cross-encoder machine learning models, and why combining them could improve the vector...

weaviate.io

Cross-Encoders - Sentence-Transformers documentation

Cross-Encoder achieve higher performance than Bi-Encoders, however, they do not scale well for large datasets. Here, it...

www.sbert.net

Thank you for reading. Stay Tuned!

Hybrid Search Cross Encoder Information Retrieval Elasticsearch Search

More from the list: "NLP"

Curated by Himanshu Birla

Jon Gi... in Towards Data ...

Characteristics of Word Embeddings

✦ · 11 min read · Sep 4, 2021

Jon Gi... in Towards Data ...

The Word2vec Hyperparameters

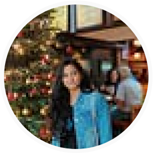
✦ · 6 min read · Sep 3, 2021

Jon Gi... in

The Word2ve

✦ · 15 min rea

View list



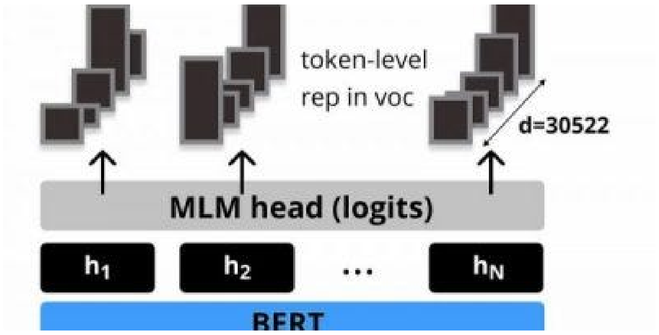
Written by Sowmiya Jaganathan

31 Followers

Following

Senior NLP Engineer - Search & AI

More from Sowmiya Jaganathan



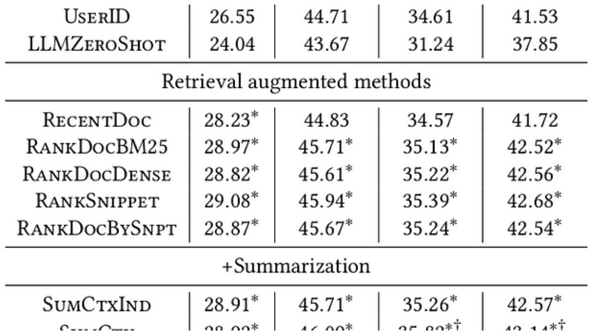
Sowmiya Jaganathan

Hybrid Search: SPLADE (Sparse Encoder)Neural retrieval models

Neural retrieval models

5 min read · Jul 9

12



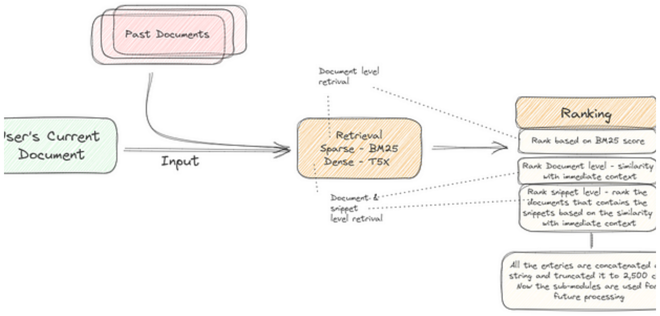
UserID	26.55	44.71	34.61	41.53
LLMZeroShot	24.04	43.67	31.24	37.85
Retrieval augmented methods				
RecentDoc	28.23*	44.83	34.57	41.72
RankDocBM25	28.97*	45.71*	35.13*	42.52*
RankDocDense	28.82*	45.61*	35.22*	42.56*
RankSnippet	29.08*	45.94*	35.39*	42.68*
RankDocBySnpt	28.87*	45.67*	35.24*	42.54*
+Summarization				
SumCtxInd	28.91*	45.71*	35.26*	42.57*

Sowmiya Jaganathan

Personalized text generation -Part II

In our previous blog post, we explored the multistage framework for generating…

3 min read · Sep 18


Past Documents
ser's Current Document
Input
Document level retrieval
Retrieval Sparse - BM25 Dense - T5X
Document & snippet level retrieval
Ranking
Rank based on BM25 score
Rank Document level - similarity with immediate context
Rank snippet level - rank the documents that contains the snippets based on the similarity with immediate context
All the enteries are concatenated string and truncated it to 2,500 Now the sub-modules are used for future processing

Sowmiya Jaganathan

Personalized text generation -Part I

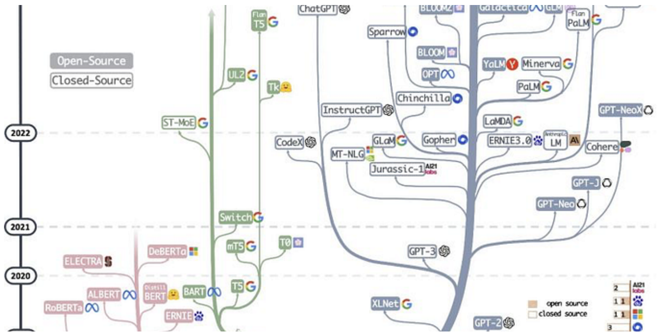
LLMs have made searching better by understanding context and generating quer…

6 min read · Sep 10

5 1

See all from Sowmiya Jaganathan

Recommended from Medium



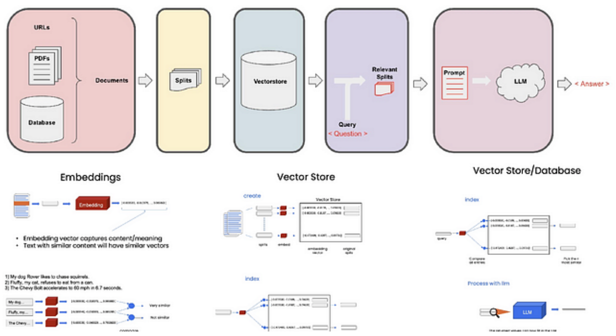
Haifeng Li

A Tutorial on LLM

Generative artificial intelligence (GenAI), especially ChatGPT, captures everyone's…

15 min read · Sep 14

372



TeeTracker

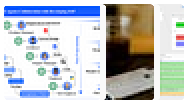
Chat with your PDF (Streamlit Demo)

Conversation with specific files

4 min read · Sep 15

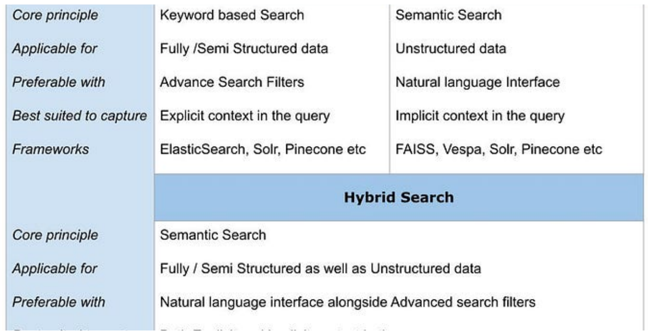
56

Lists



Natural Language Processing

669 stories · 283 saves



Core principle	Keyword based Search	Semantic Search
Applicable for	Fully /Semi Structured data	Unstructured data
Preferable with	Advance Search Filters	Natural language Interface
Best suited to capture	Explicit context in the query	Implicit context in the query
Frameworks	ElasticSearch, Solr, Pinecone etc	FAISS, Vespa, Solr, Pinecone etc
	Hybrid Search	
Core principle	Semantic Search	
Applicable for	Fully / Semi Structured as well as Unstructured data	
Preferable with	Natural language interface alongside Advanced search filters	

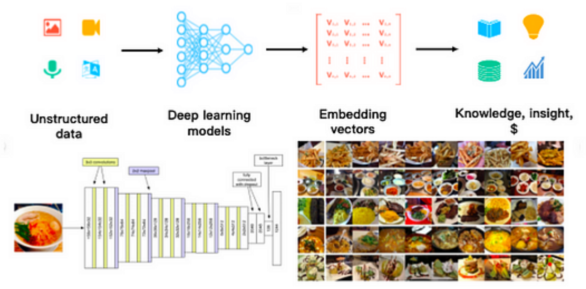
Maithri Vm

Hybrid Search — Amalgamation of Sparse and Dense vector…

— Uniting meaning of data with metadata to leverage deeper context

13 min read · May 14

16



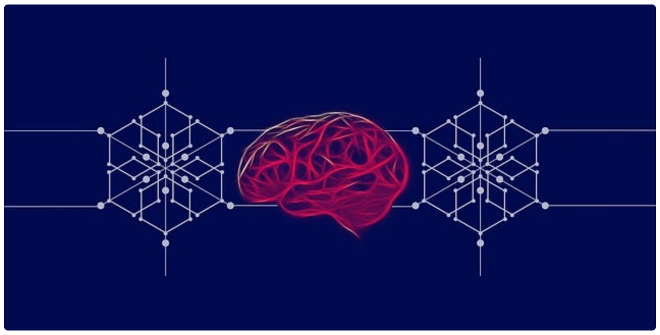
Jayita Bhattacharyya in GoPenAI

Primer on Vector Databases and Retrieval-Augmented Generation…

Vector Databases Generation (RAG) Langchain Pinecone HuggingFace Large…

9 min read · Aug 16

228 1



ai geek (wishesh)

Best Practices for Deploying Large Language Models (LLMs) in…

Large Language Models (LLMs) have revolutionized the field of natural language…

10 min read · Jun 26

David Shapiro

A Pro's Guide to Finetuning LLMs

Large language models (LLMs) like GPT-3 and Llama have shown immense promise for…

12 min read · Sep 23

100 1

283 6

See more recommendations