Pinecone

Neural Retrieval Meets Cascading Architectures

April 10th 2025

Antonio Mallia, Staff Research Scientist
Cesare Campagnano, Senior Research Scientist
Jack Pertschuk, Principal Engineer

About us

Antonio Mallia

I am currently a Staff Research Scientist at Pinecone.
Prior to this, I served as an Applied Scientist on the Artificial General Intelligence (AGI) team at Amazon.
I hold a Ph.D. from New York University, where my research focused on efficient web retrieval methodologies.

Cesare Campagnano

I am currently a Senior Research Scientist at Pinecone.
I earned my Ph.D. from Sapienza University of Rome, under Gabriele Tolomei and Fabrizio Silvestri, during which I also interned at Amazon.
Afterward, I held a PostDoc position at the same institution. My research interests revolve around Large Language Models and Information Retrieval.

Special Guest!

Jack Pertschuk

Jack is a founding engineer and manager of the Applied Research and Algorithms team at Pinecone. Prior to Pinecone, Jack was a founder of SidekickQA and creator of the NBoost open source neural ranking engine.
He is an active member of the NYC Rust community and is passionate about solving problems at the intersection of ML and databases.

About Pinecone

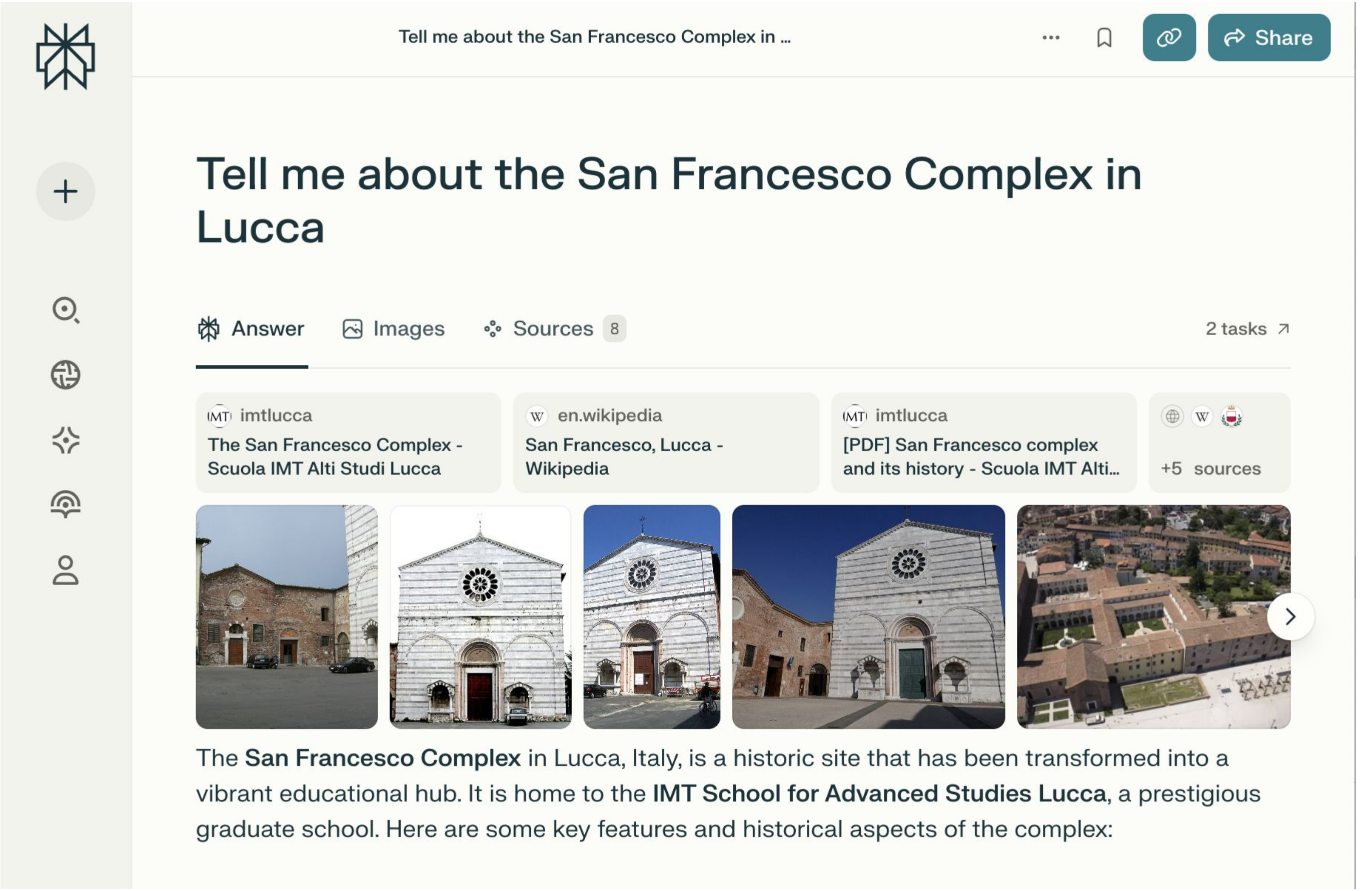
- Started as a vector database in 2019
- Founded by Edo Liberty
- Today, Pinecone's mission is to make AI knowledgeable
- The product offering has recently expanded to include new IR features
- Go and check it out: https://www.pinecone.io/

Agenda

- Search & RAG
- Cascading retrieval
- Sparse & dense
- Multi-vector
- Cascading Reranking
- Demo

“Knowledgeable AI requires better retrieval, not just better LLMs”

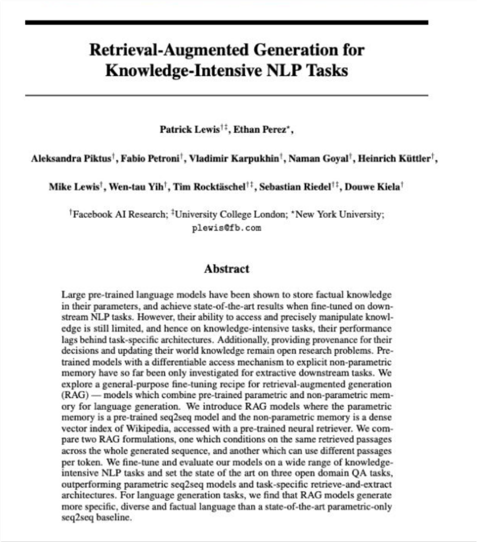
Modern Search



Retrieval-Augmented Generation (RAG)

Combines retrieval of external information with generation capabilities of language models.

- First introduced in “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks” (Lewis et al., 2020)
- Incorporates up-to-date and specific information not contained within the model
- Reduces hallucinations by grounding responses in real data
- Include citations to the original sources, enhancing credibility and traceability
- Allows models to leverage vast external databases without the need for extensive retraining



Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis[†‡], Ethan Perez[*],

Aleksandra Piktus[†], Fabio Petroni[†], Vladimir Karpukhin[†], Naman Goyal[†], Heinrich Küttler[†],

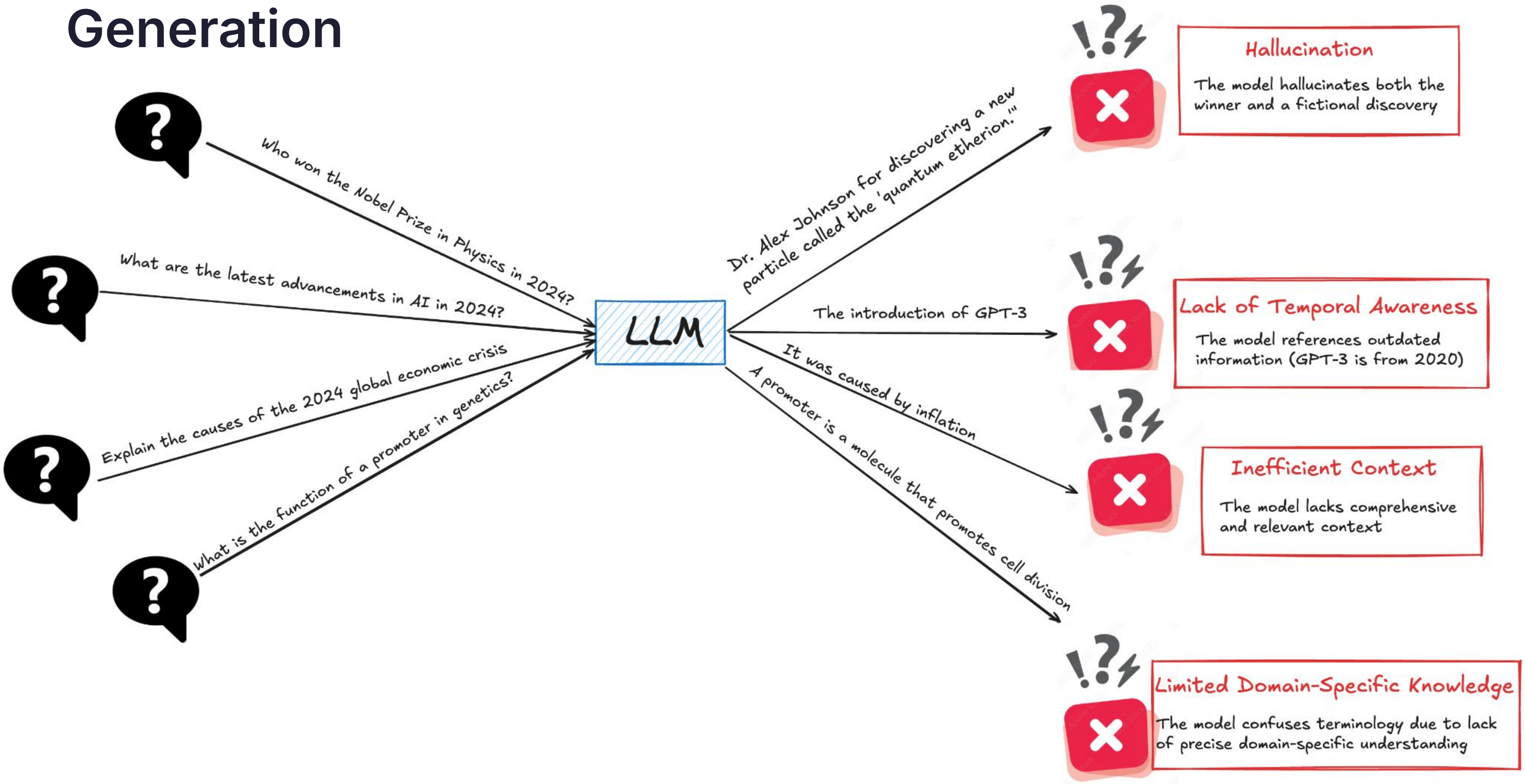
Mike Lewis[†], Wen-tau Yih[†], Tim Rocktäschel[†‡], Sebastian Riedel[†‡], Douwe Kiela[†]

[†]Facebook AI Research; [‡]University College London; [*]New York University;
plewis@fb.com

Abstract

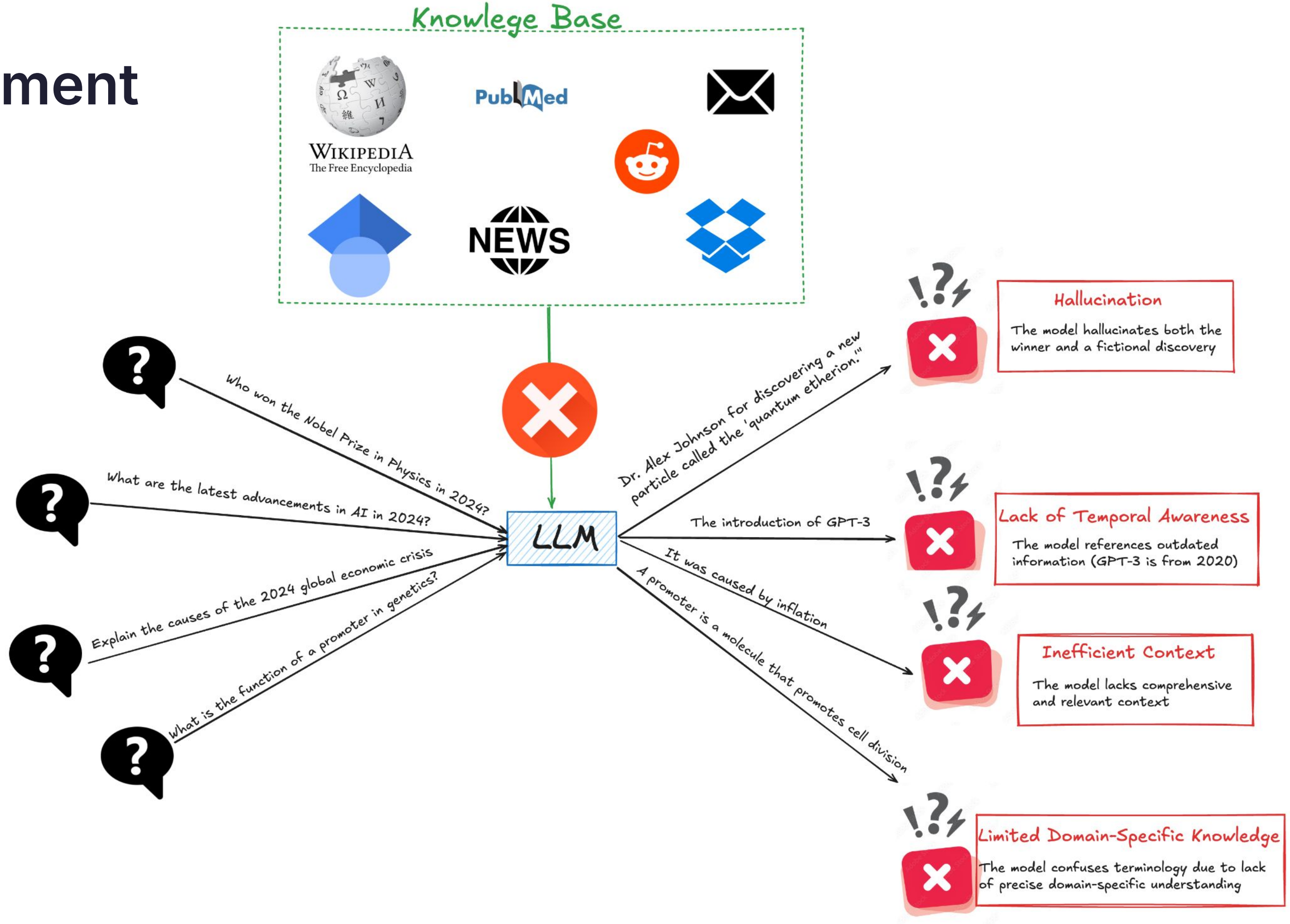
Large pre-trained language models have been shown to store factual knowledge in their parameters, and achieve state-of-the-art results when fine-tuned on downstream NLP tasks. However, their ability to access and precisely manipulate knowledge is still limited, and hence on knowledge-intensive tasks, their performance lags behind task-specific architectures. Additionally, providing provenance for their decisions and updating their world knowledge remain open research problems. Pre-trained models with a differentiable access mechanism to explicit non-parametric memory have so far been only investigated for extractive downstream tasks. We explore a general-purpose fine-tuning recipe for retrieval-augmented generation (RAG) — models which combine pre-trained parametric and non-parametric memory for language generation. We introduce RAG models where the parametric memory is a pre-trained seq2seq model and the non-parametric memory is a dense vector index of Wikipedia, accessed with a pre-trained neural retriever. We compare two RAG formulations, one which conditions on the same retrieved passages across the whole generated sequence, and another which can use different passages per token. We fine-tune and evaluate our models on a wide range of knowledge-intensive NLP tasks and set the state of the art on three open domain QA tasks, outperforming parametric seq2seq models and task-specific retrieve-and-extract architectures. For language generation tasks, we find that RAG models generate more specific, diverse and factual language than a state-of-the-art parametric-only seq2seq baseline.

Generation



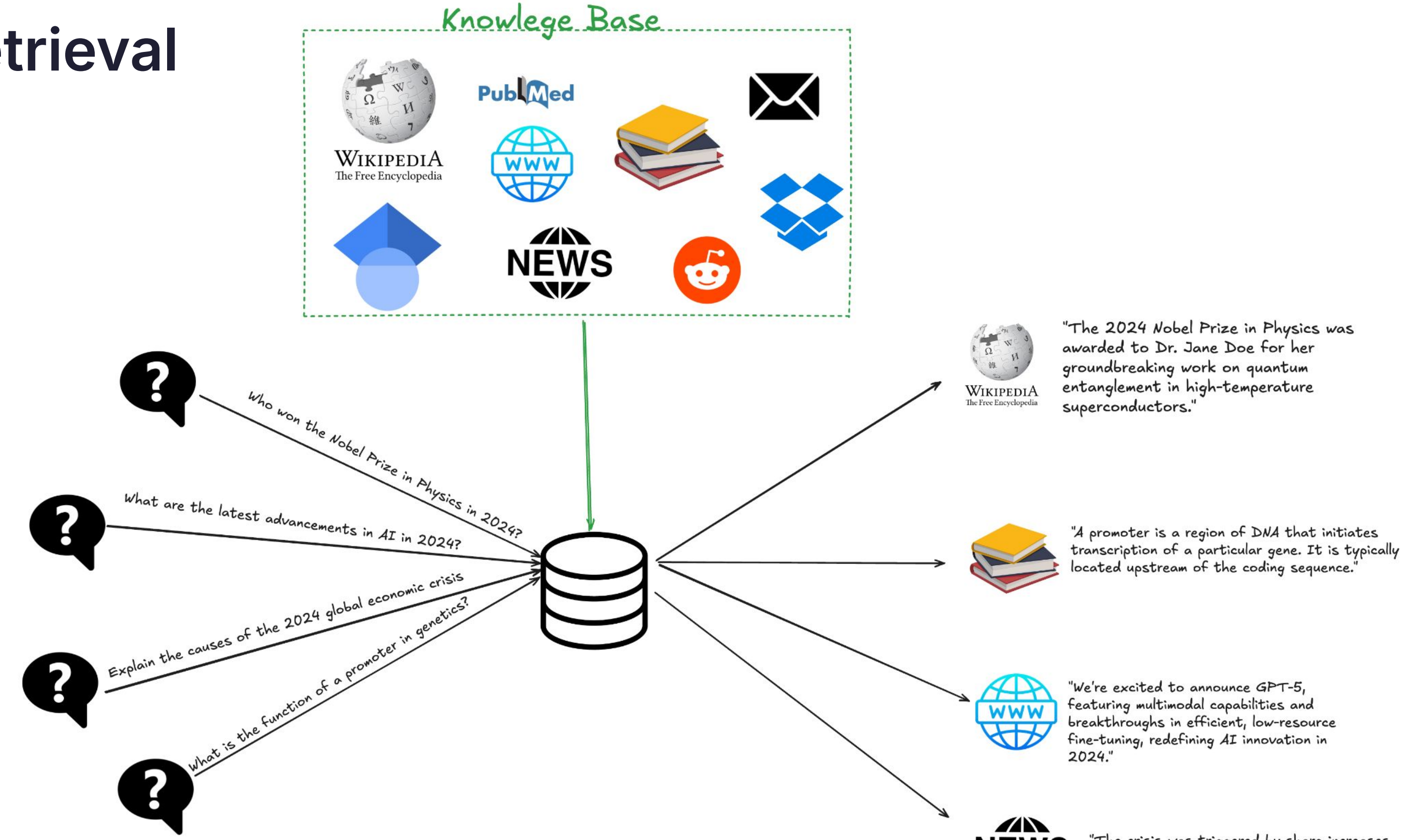
Who won the Nobel Prize in Physics in 2024?
What are the latest advancements in AI in 2024?
Explain the causes of the 2024 global economic crisis
What is the function of a promoter in genetics?
LLM
Dr. Alex Johnson for discovering a new particle called the 'quantum etherion.'"
The introduction of GPT-3
It was caused by inflation
A promoter is a molecule that promotes cell division
Hallucination
The model hallucinates both the winner and a fictional discovery
Lack of Temporal Awareness
The model references outdated information (GPT-3 is from 2020)
Inefficient Context
The model lacks comprehensive and relevant context
Limited Domain-Specific Knowledge
The model confuses terminology due to lack of precise domain-specific understanding

Augment



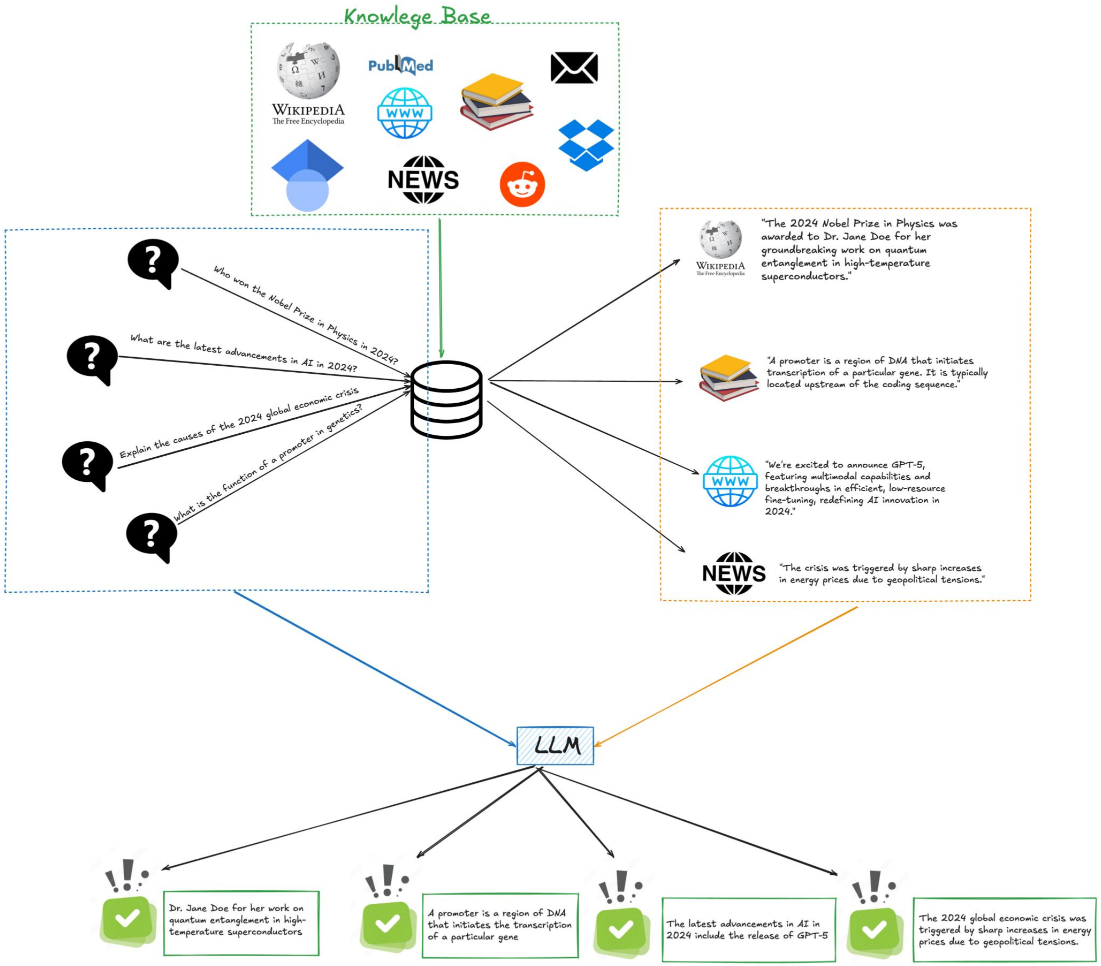
Knowlege Base
Wikipedia
The Free Encyclopedia
PubMed
NEWS
Who won the Nobel Prize in Physics in 2024?
What are the latest advancements in AI in 2024?
Explain the causes of the 2024 global economic crisis
What is the function of a promoter in genetics?
LLM
Dr. Alex Johnson for discovering a new particle called the 'quantum etherion."
The introduction of GPT-3
It was caused by inflation
A promoter is a molecule that promotes cell division
Hallucination
The model hallucinates both the winner and a fictional discovery
Lack of Temporal Awareness
The model references outdated information (GPT-3 is from 2020)
Inefficient Context
The model lacks comprehensive and relevant context
Limited Domain-Specific Knowledge
The model confuses terminology due to lack of precise domain-specific understanding

Retrieval



Knowlege Base
PubMed
NEWS
Who won the Nobel Prize in Physics in 2024?
What are the latest advancements in AI in 2024?
Explain the causes of the 2024 global economic crisis
What is the function of a promoter in genetics?
"The 2024 Nobel Prize in Physics was awarded to Dr. Jane Doe for her groundbreaking work on quantum entanglement in high-temperature superconductors."
"A promoter is a region of DNA that initiates transcription of a particular gene. It is typically located upstream of the coding sequence."
"We're excited to announce GPT-5, featuring multimodal capabilities and breakthroughs in efficient, low-resource fine-tuning, redefining AI innovation in 2024."
NEWS
"The crisis was triggered by sharp increases in energy prices due to geopolitical tensions."

RAG

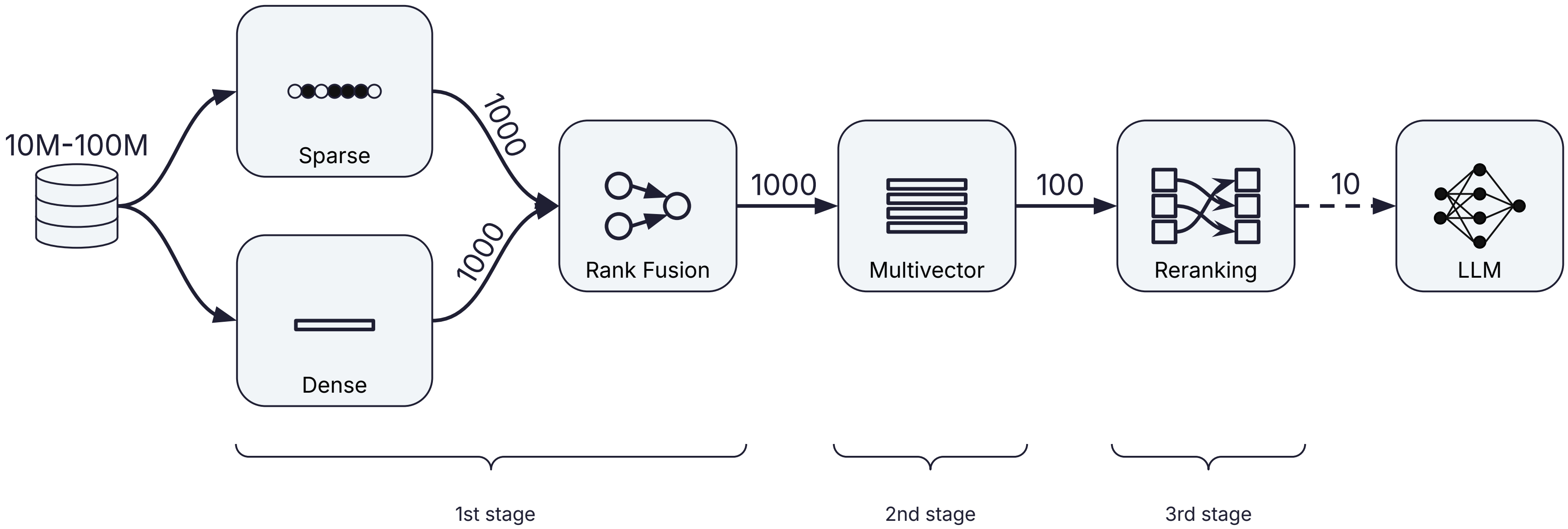


Knowlege Base
WIKIPEDIA The Free Encyclopedia
PubMed
WWW
NEWS
Who won the Nobel Prize in Physics in 2024?
What are the latest advancements in AI in 2024?
Explain the causes of the 2024 global economic crisis
What is the function of a promoter in genetics?
"The 2024 Nobel Prize in Physics was awarded to Dr. Jane Doe for her groundbreaking work on quantum entanglement in high-temperature superconductors."
"A promoter is a region of DNA that initiates transcription of a particular gene. It is typically located upstream of the coding sequence."
"We're excited to announce GPT-5, featuring multimodal capabilities and breakthroughs in efficient, low-resource fine-tuning, redefining AI innovation in 2024."
"The crisis was triggered by sharp increases in energy prices due to geopolitical tensions."
LLM
Dr. Jane Doe for her work on quantum entanglement in high-temperature superconductors
A promoter is a region of DNA that initiates the transcription of a particular gene
The latest advancements in AI in 2024 include the release of GPT-5
The 2024 global economic crisis was triggered by sharp increases in energy prices due to geopolitical tensions.

Cascading retrieval

Cascading Retrieval: combining sparse, dense, & rerank

Using two indexes rather than a hybrid index increases **recall** while the reranker ensures **accuracy** of the final results.



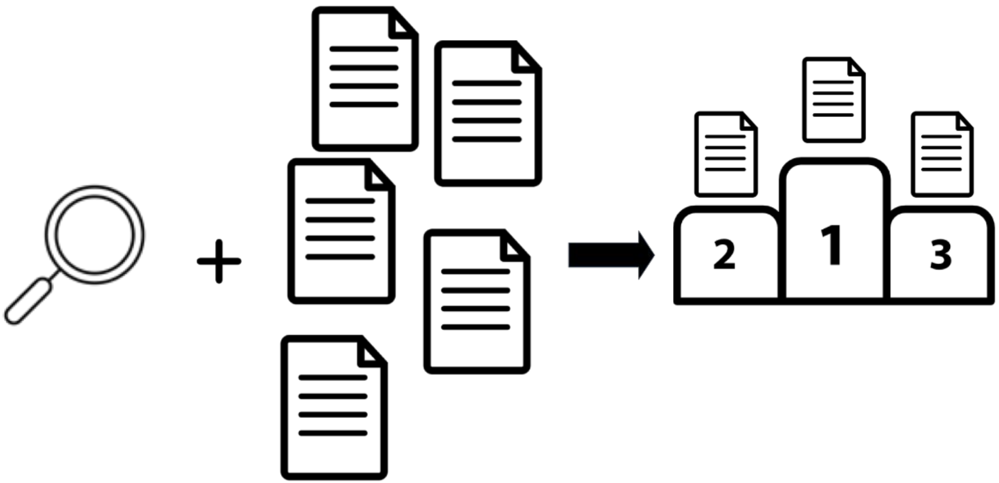
Sparse & Dense

Search in a Nutshell

Given: a query and a collection of documents

Return: a ranked list of k documents

Maximizing: a metric of interest



The scoring function assigns a **score** to each document.

It allows to return only the **top-k** documents with highest scores.

Popular functions:

- Cosine similarity
- Tf-idf
- BM25
- Language models

Vocabulary Mismatch

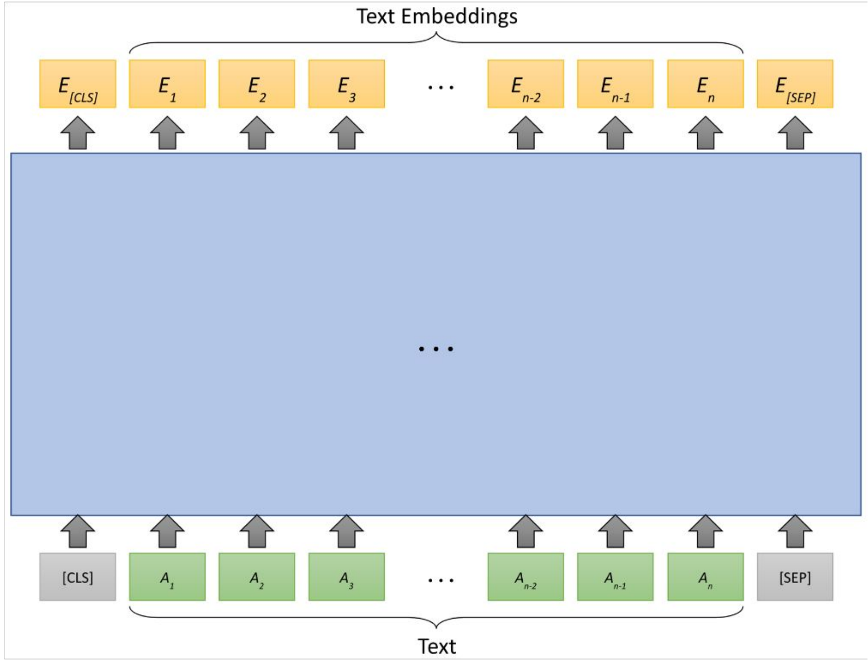
How many people live in Rome?

Rome's population is 4.3 million

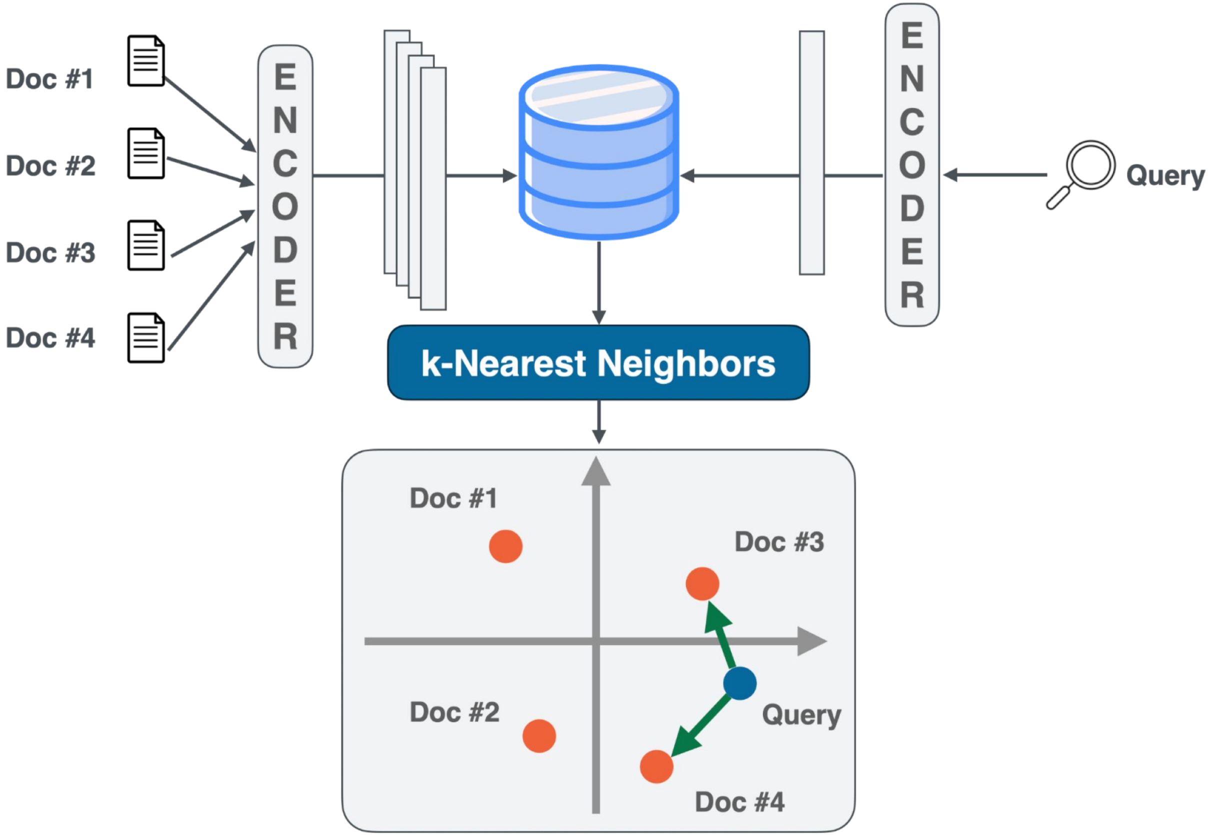
Hundreds of people queuing for live music in Rome

Dense Retrieval

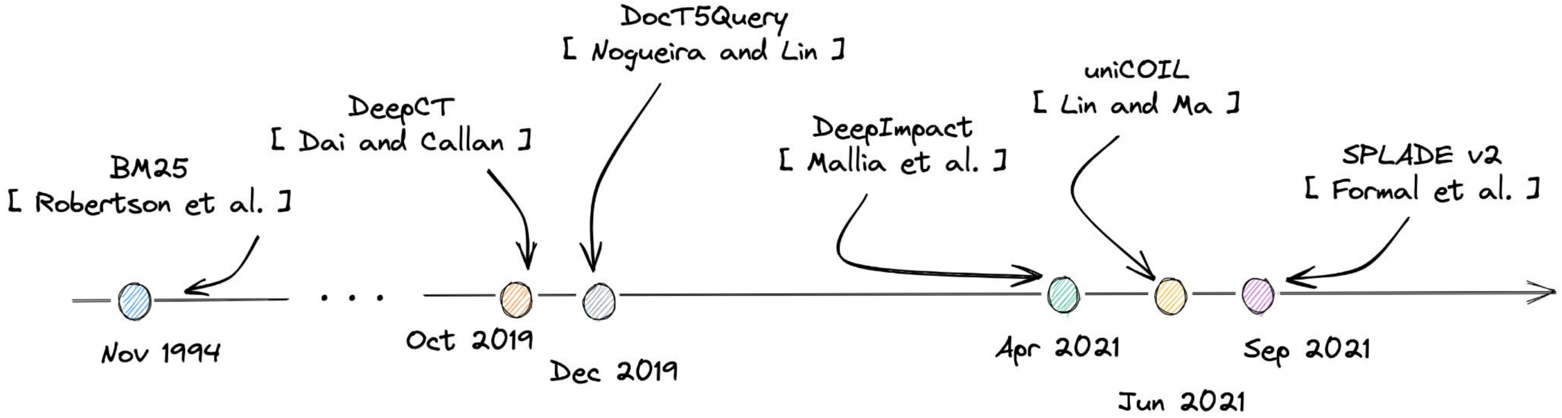
- An alternative to BM25 for first-stage retrieval
- **Dense Embeddings**: Queries and documents are encoded into dense vector representations in a shared embedding space.
- **Semantic Matching**: DR captures contextual and semantic relevance beyond keyword overlap, improving search quality for complex or ambiguous queries.
- **Dual Encoder Architecture**: Queries and documents are encoded independently for efficient offline indexing.
- Dense retrieval relies on nearest neighbor approaches



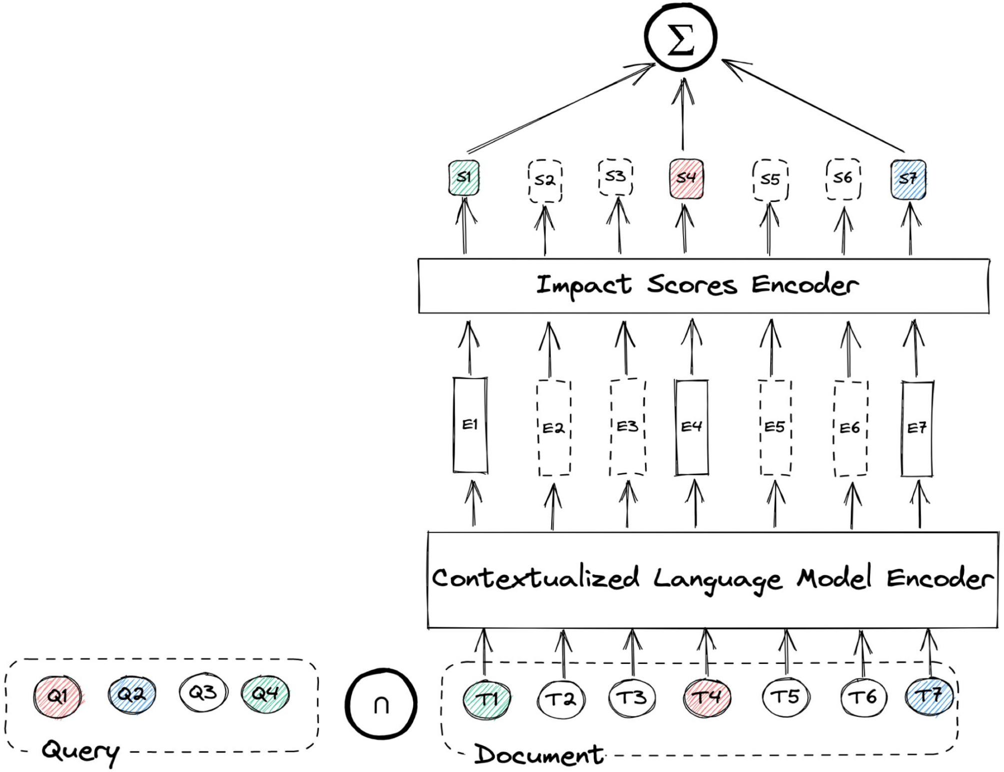
k-Nearest Neighbour (kNN)



Sparse timeline

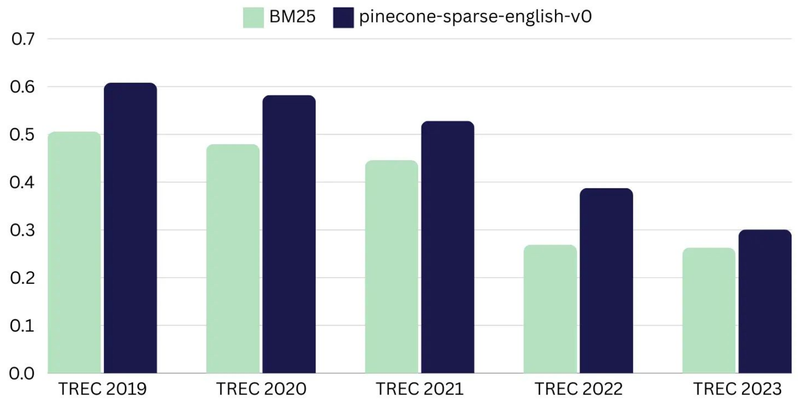


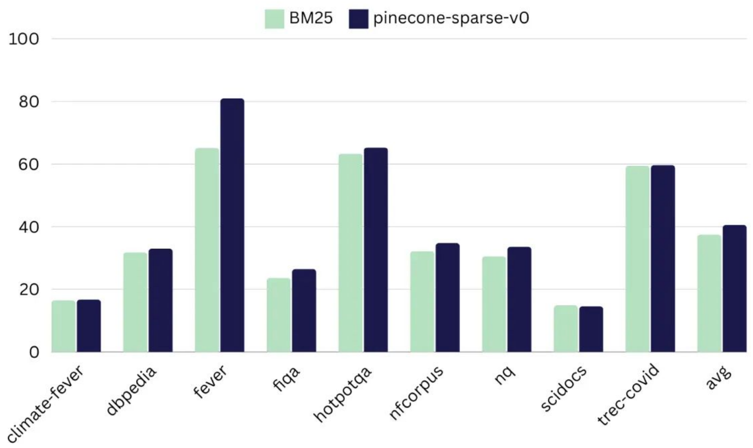
Learned Sparse Retrieval



Benchmarking pinecone-sparse-english-v0

23% better NDCG@10 on TREC Deep Learning and 8% on BEIR than BM25





Pinecone's Sparse Model

Learned sparse methods estimate the importance of keywords using context, in contrast BM25 relies on corpus term frequency

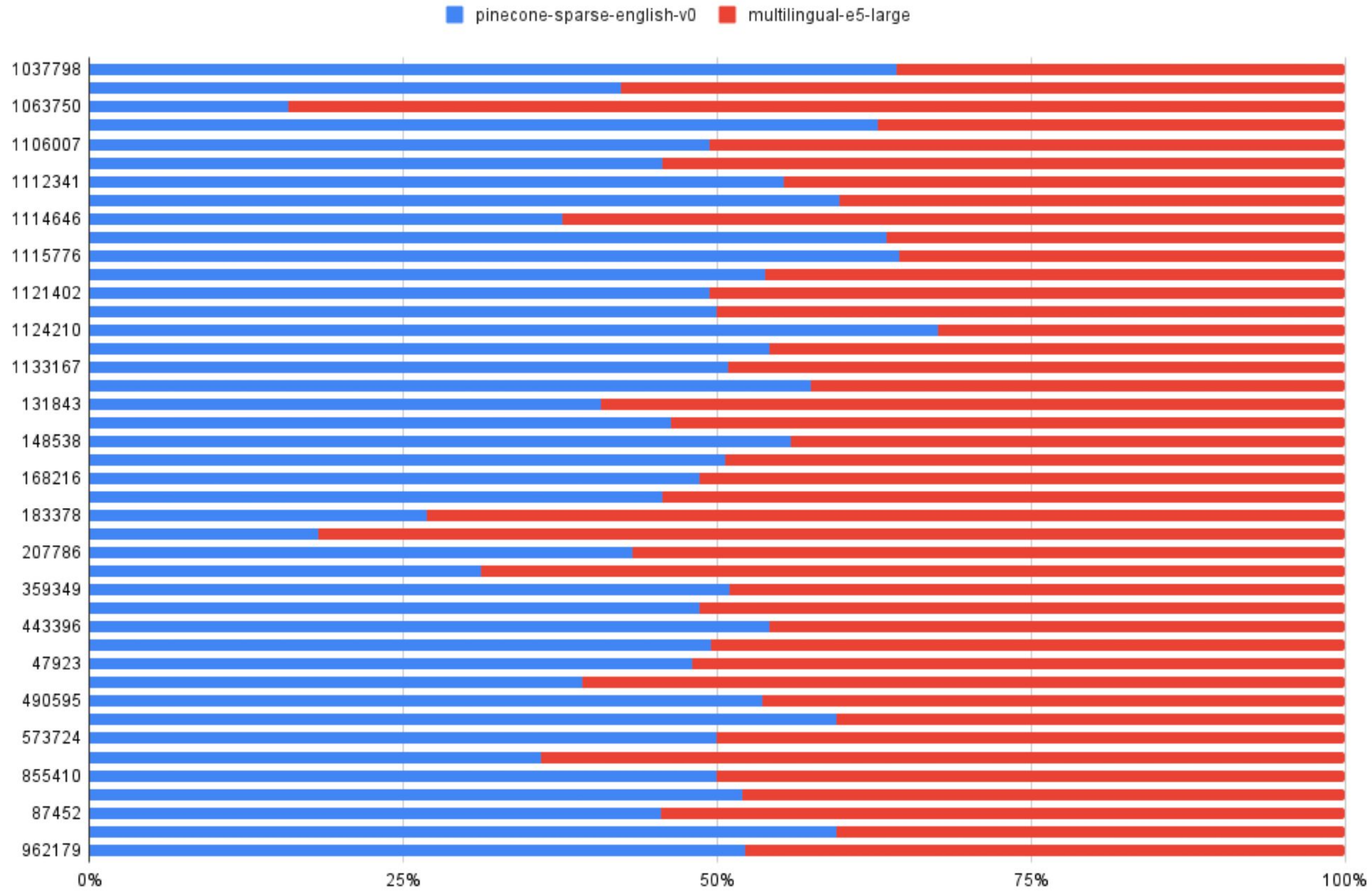
pinecone-sparse-v0	BM25	Example
2.46	0.61	**Does** are the females in the deer family of mammals, individually called a doe.
0.20	0.62	**Does** anyone else want to come to the movies with us?

Fusion

Semantic Textual Similarity

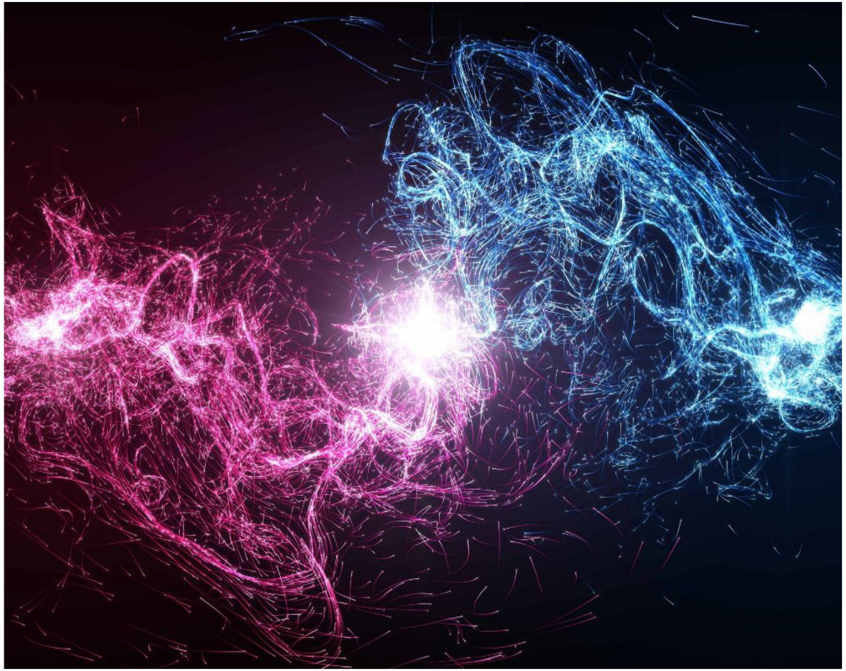
Source: https://ai.googleblog.com/2018/05/advances-in-semantic-textual-similarity.html

Dense vs. Sparse Retrieval



Rank fusion

Fusion for Information Retrieval is the the process of combining multiple sources of information to produce a single result list in response to a query.



CombSUM

CombSUM is a simple and popular rank fusion method used in Information Retrieval (IR) to combine the scores of documents retrieved from multiple retrieval models or query representations. It is part of the **CombMNZ family** of score combination techniques introduced by Fox and Shaw (1994).

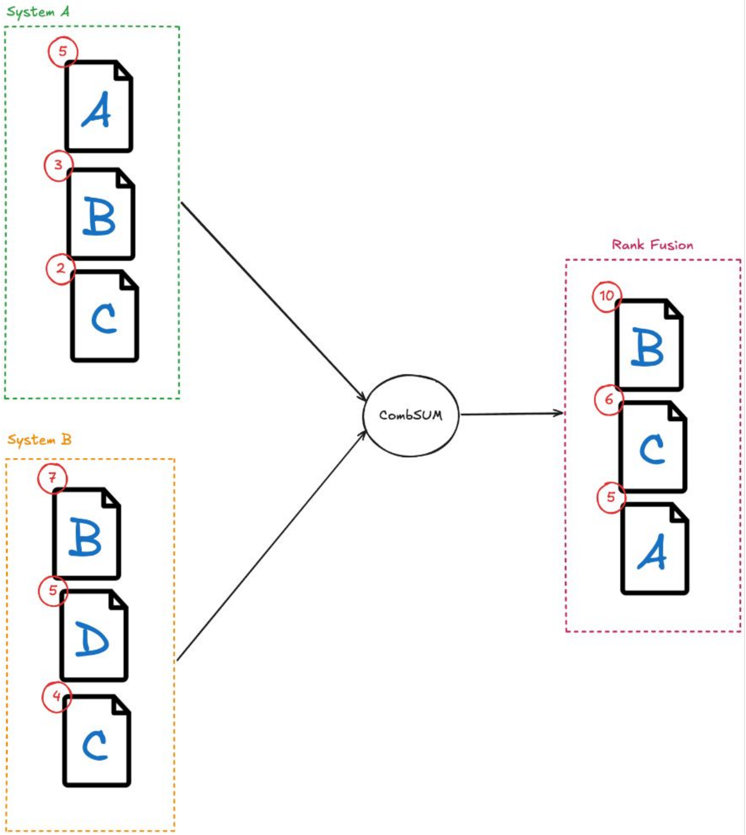
How CombSUM Works:

1. **Input**: Several ranked lists from different retrieval systems, each assigning a relevance score to the documents.
2. **For Each Document**:
   - Retrieve the scores assigned by each individual system.
   - Compute the **sum of these scores** across all systems.
3. **Ranking**:
   - Sort the documents in descending order of the aggregated scores.

If a document d has scores $S_1(d), S_2(d), \ldots, S_n(d)$ from n systems, its combined score using CombSUM is:

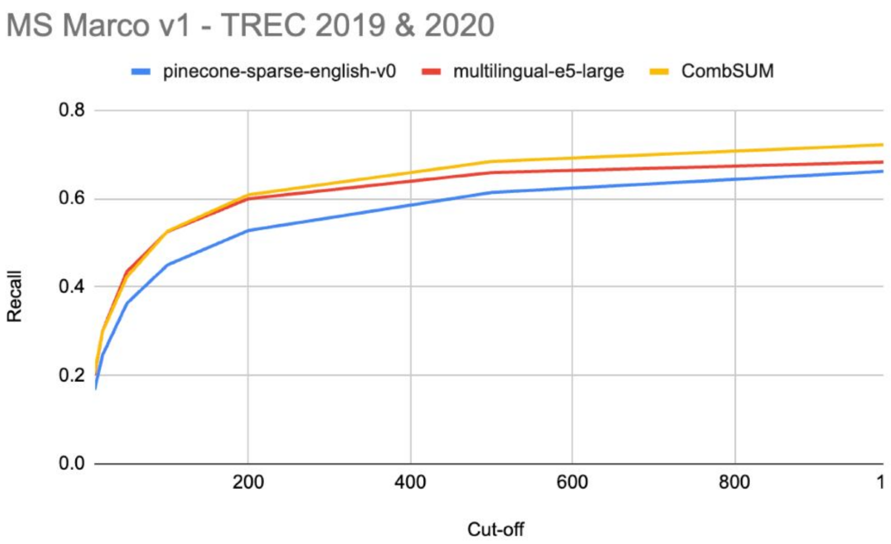
$$\text{Score}_{\text{CombSUM}}(d) = \sum_{i=1}^{n} S_i(d)$$

How it works



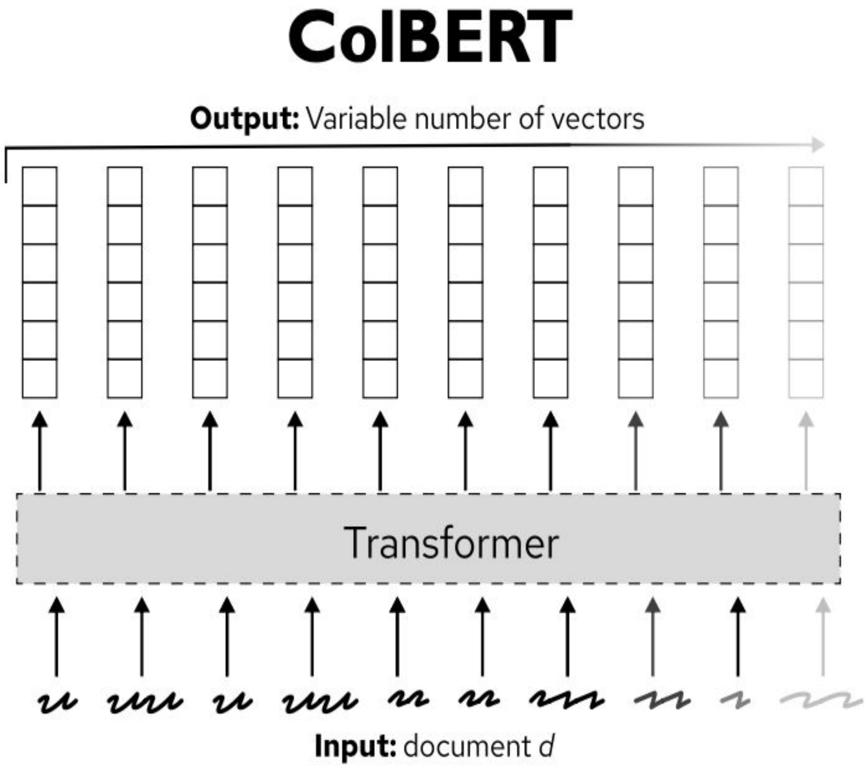
Impact: Doc B moves to the top after CombSum due to its strong performance in System B, showcasing how aggregation captures complementary strengths.

Improved Recall



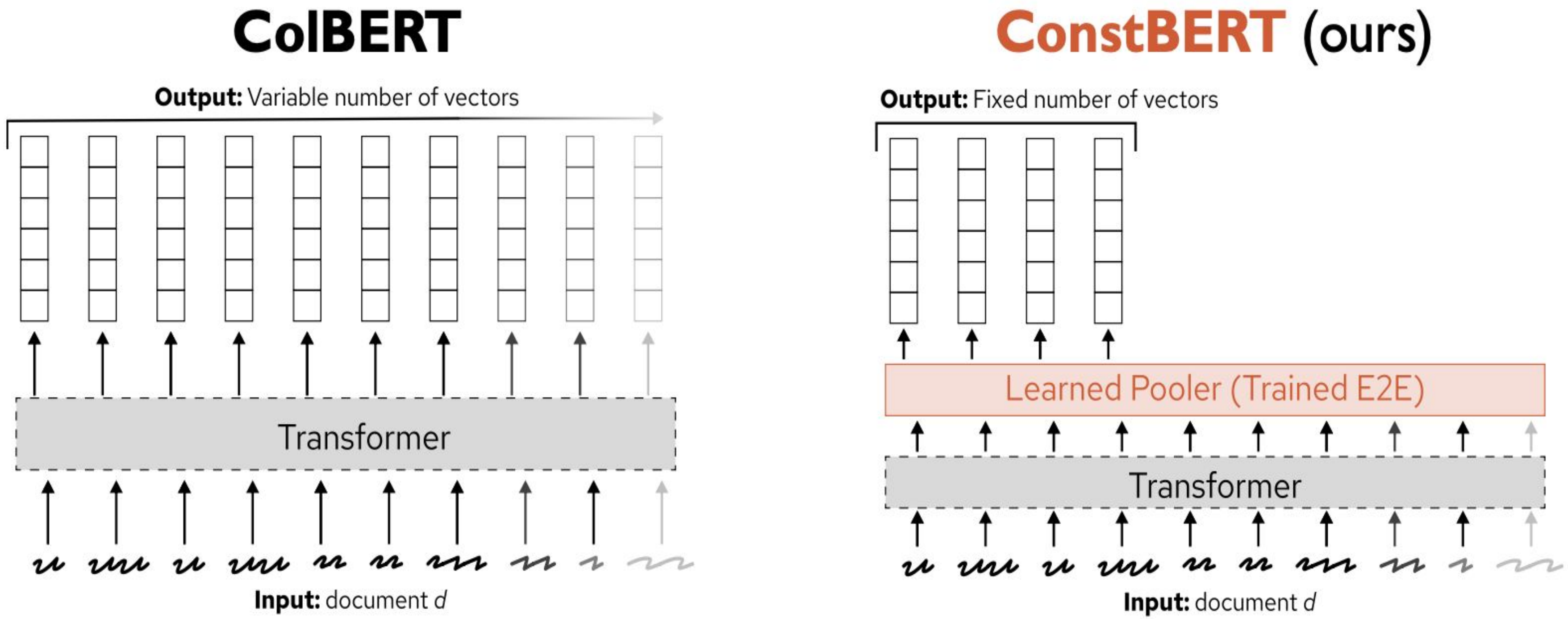
Multi-Vector Reranking

Late interaction ranking with Multi-vector models

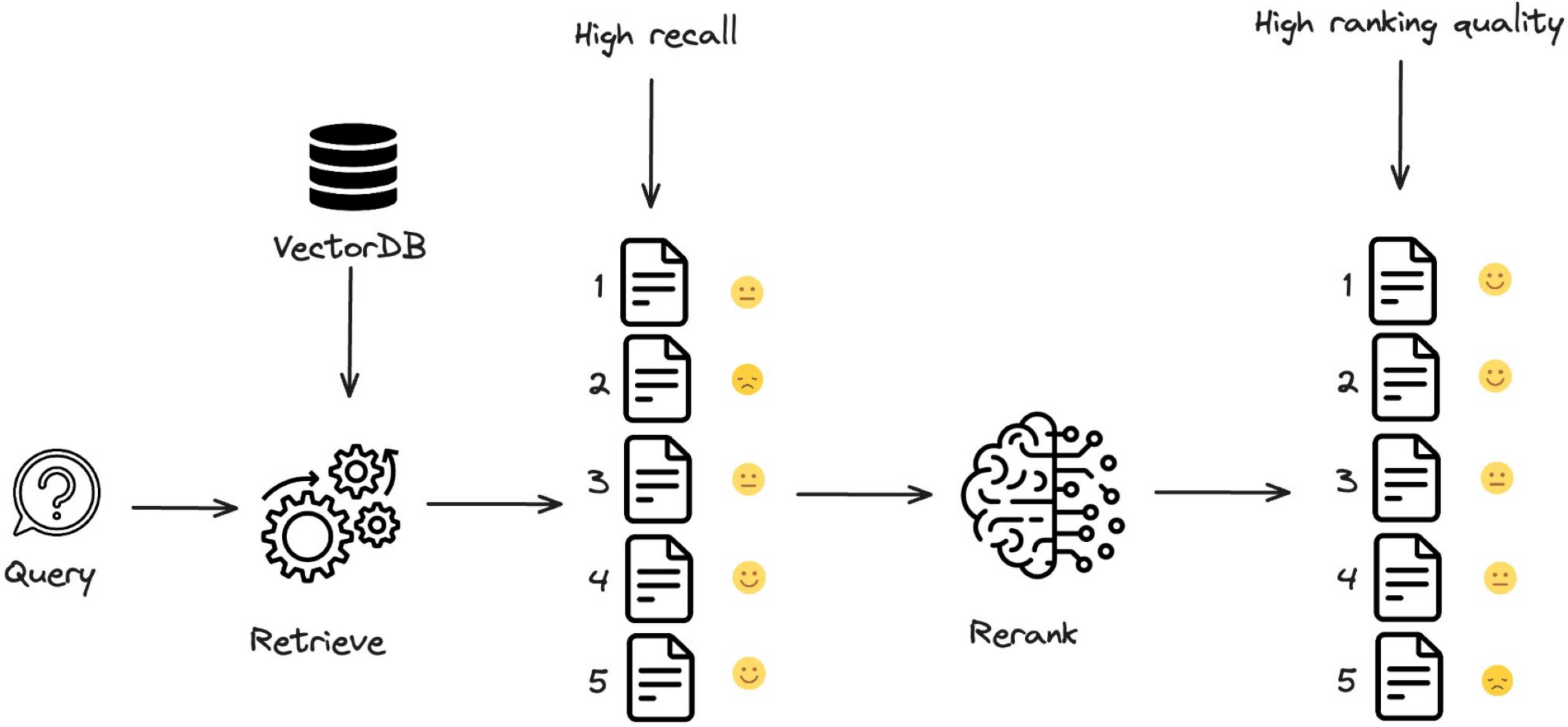


- ColBERT's retrieval component is **not optimized** during training
- Late interaction is **more expensive** than Bi-Encoder models
- In a multi-vector system, the first stage of retrieval employs a **single vector interactions**
- The second stage perform **full score computation** with multi-vector representations

Multi-vector



What is Reranking?



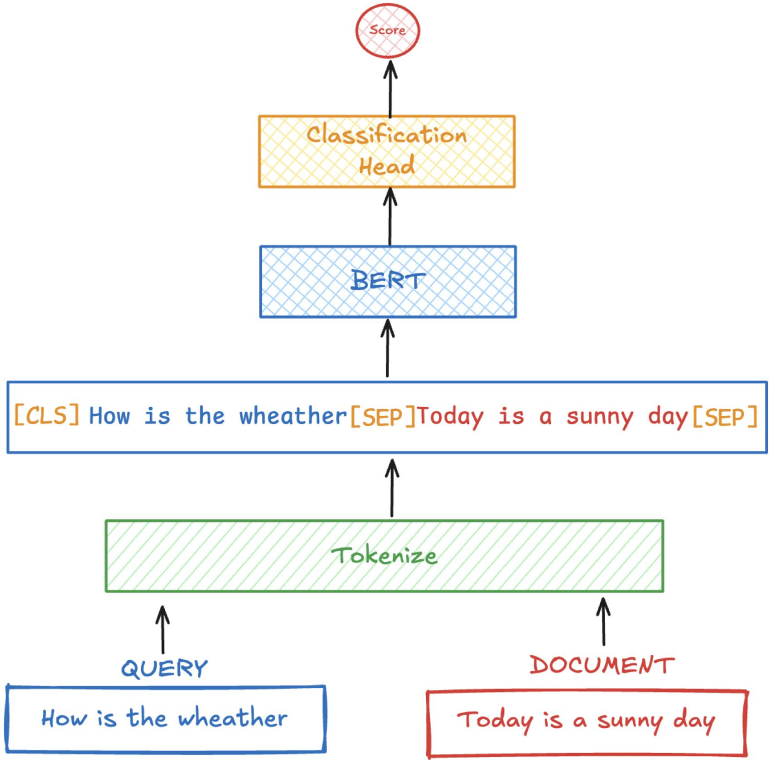
ConstBERT for Reranking

ConstBERT as a reranking model instead of employing it in an end-to-end retrieval system. Simple and accurate!

Model	NDCG@10
ColBERT	70.1
E5	71.1
ConstBERT	74.5
E5 + ConstBERT (reranking)	79.0

Cross-encoders for reranking

Cross-encoder as a reranker



Joint Encoding: Both the query and the document are tokenized and concatenated with a [SEP] token, which separates the two inputs to inform the model of their distinct roles.

Contextual Representation: BERT processes the concatenated input to generate contextual embeddings, capturing relationships between query and document tokens.

Classification Head: The [CLS] token embedding, which represents the combined query-document pair, is passed to a classification head to compute a relevance score.

Relevance Scoring: The final score determines how well the document matches the query, enabling accurate reranking of candidate documents in search.

Large cross-encoder architectures

✅ High accuracy

But...

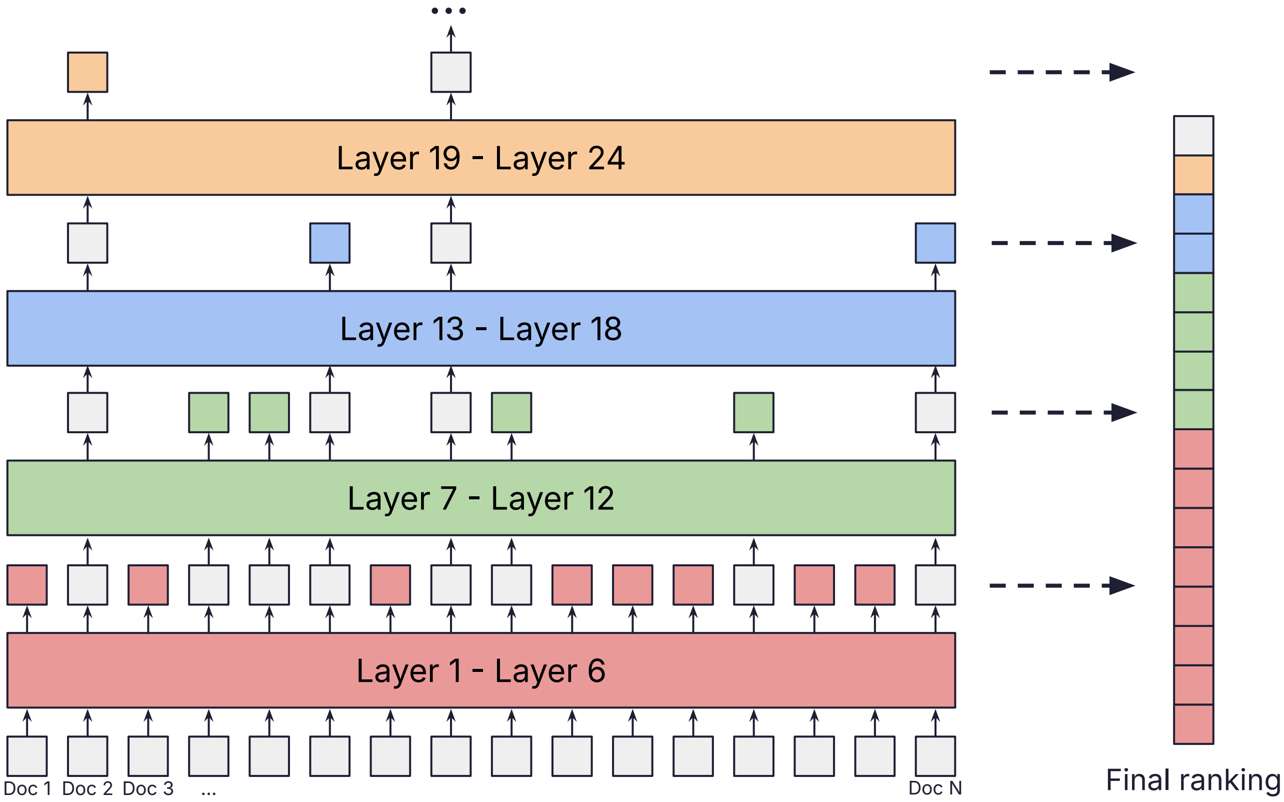
❌ Cannot be pre-computed

❌ Slow

❌ Not suitable for retrieval

❌ Does not scale

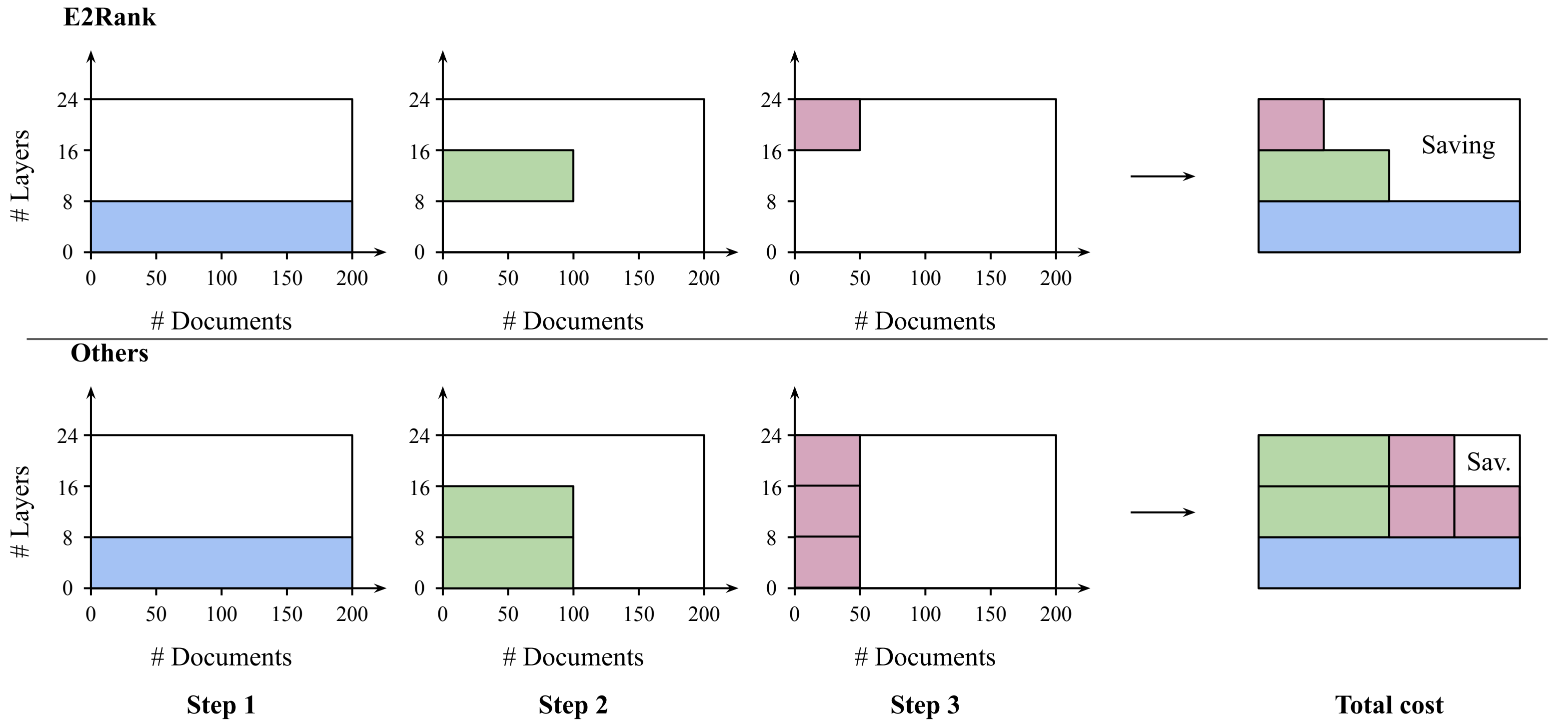
Computation reuse in E2Rank



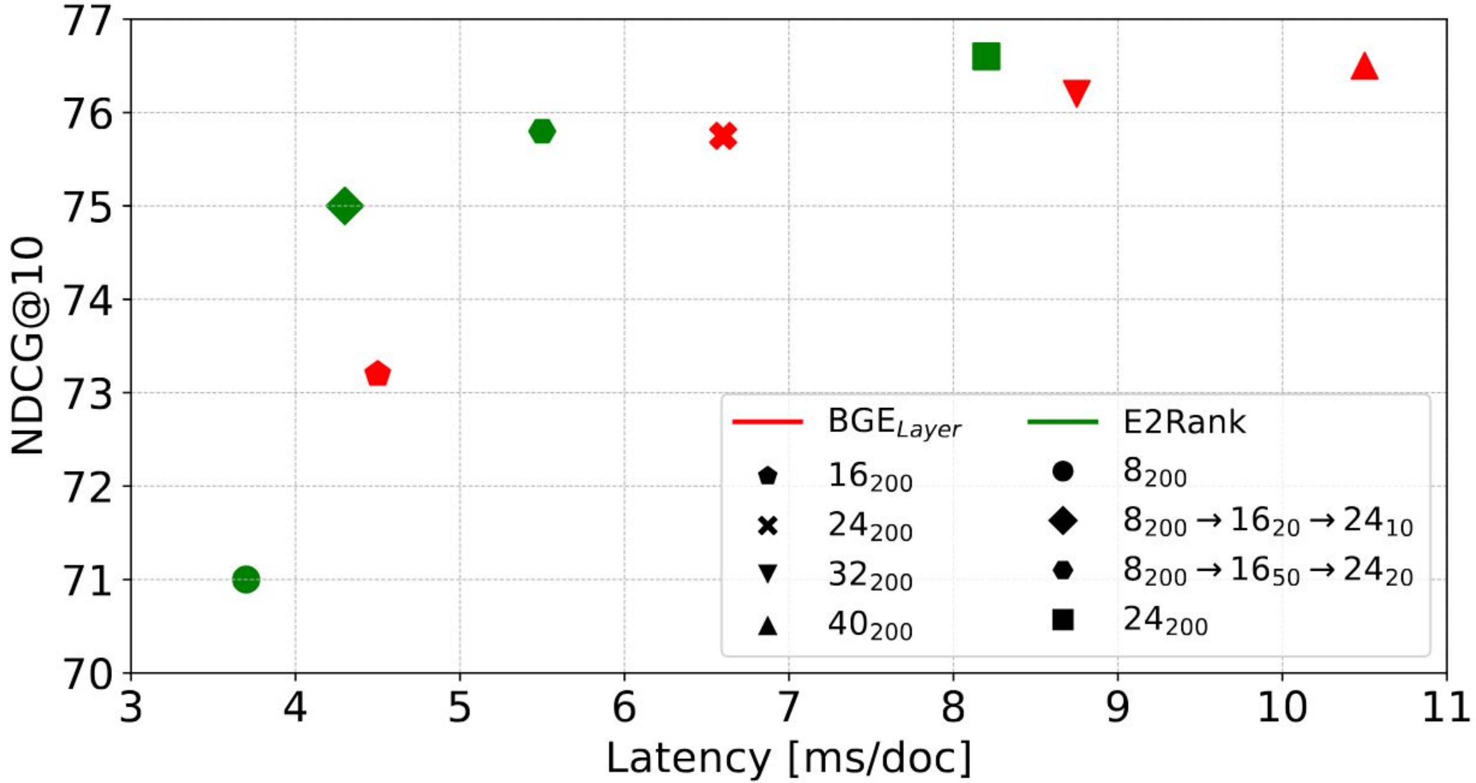
Most promising docs survive across layers:

- More efficient
- Negligible loss in accuracy

Multi-step Reranking



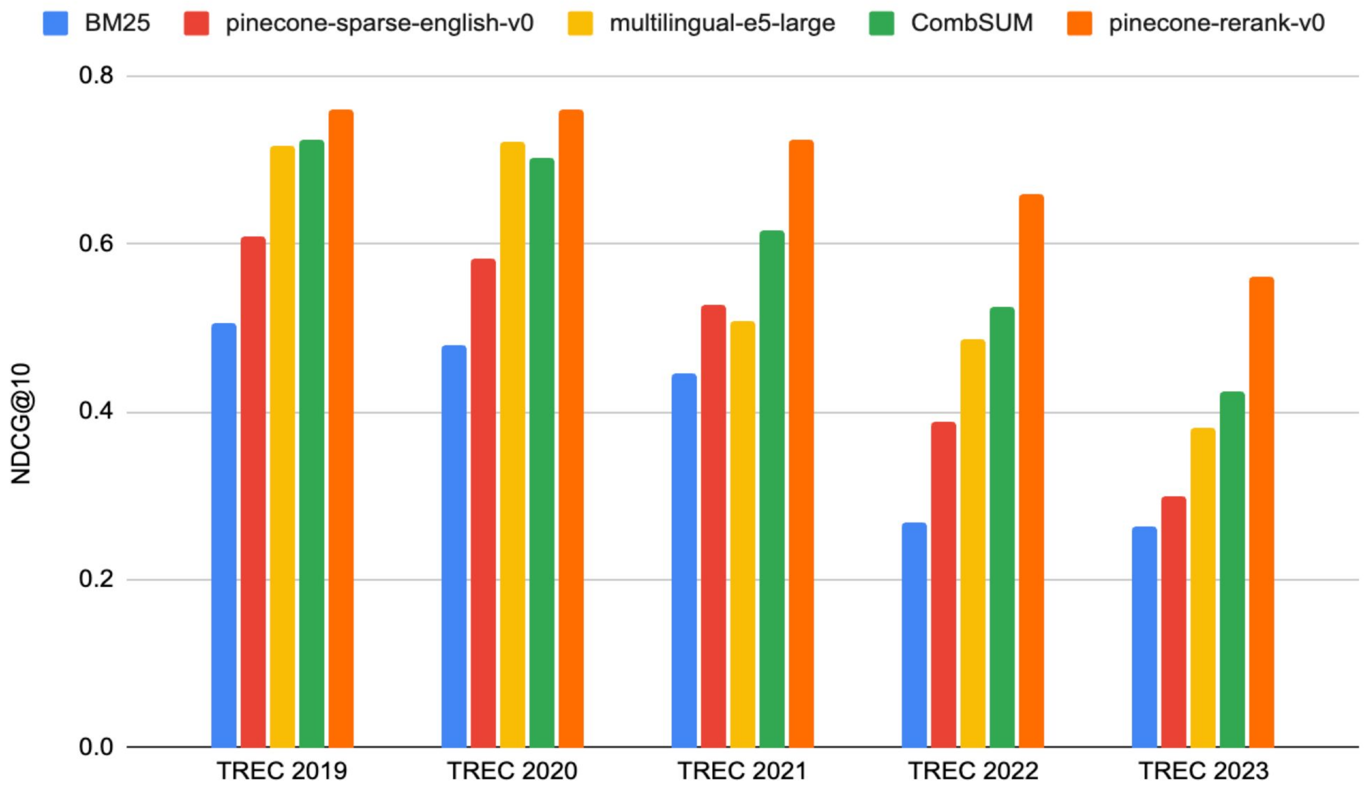
Efficiency-effectiveness tradeoffs



A comparison

	Bi-encoder	Late interaction	Cross-encoder
Retrieval-ready	✅	⚠️	❌
Precomputable	✅	✅	❌
Efficient at scale	✅	⚠️	❌
High accuracy	❌	⚠️	✅
Use as a reranker	⚠️	✅	✅

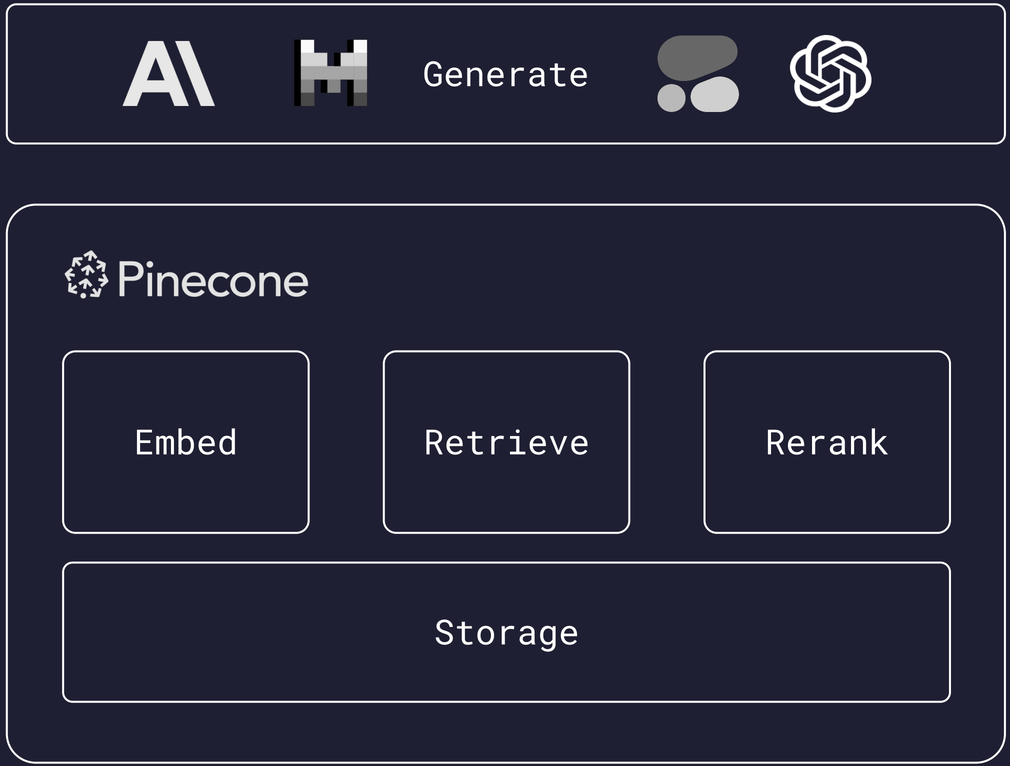
Cross-encoder as a reranker



Thanks!

Any questions?

A composable platform for knowledgeable AI



DEMO

Lost in the Middle

When using large context windows, LLMs struggle to attend to the relevant content

The power of noise (Cesare)

Related but not relevant documents in retrieved results harm generated answers

Comparison

Problems	ColBERT	ConstBERT (32)
Data size	✘ N ∗ 128 ∗ 4 Byte With N up to 512, which means ~78	✓ 32 ∗ 128 ∗ 4 Byte = 16kb
Hypothesis of independent index	✘ Vectors cannot be concatenated into a single vector (as they would have different sizes)	✓ (vectors can be concatenated into a single, fixed size vector, e.g. 32x128 = 4096)
Number of vectors	✘ Larger amount of vectors (on average), including a bad worst case scenario	✓ Small and predictable (e.g. with 256 tokens, 8x reduction)
OS paging	✘ Irregular data patterns and inefficient pre-fetching	✓ Efficient memory alignment with OS-level paging and pre-fetching