Sparse Retrieval in the Age of RAG

Antonio Mallia

LiveRAG Challenge @ SIGIR'2025

July 17, 2025

About me

me@antoniomallia.it

www.antoniomallia.it

@antonio_mallia

amallia

in/antoniomallia

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.

Pinecone

Founded by Edo Liberty

Created the vector database category

Pinecone's mission is to make AI knowledgeable

Pinecone is the leading vector database for building accurate and performant AI applications at scale in production.

Go and check it out: https://www.pinecone.io/

Agenda

Traditional IR vs. RAG systems

Sparse Retrieval in LiveRAG

Advantages of Sparse

Not a replacement for Dense: hybrid

Future: Learned Sparse Retrieval for RAG

The Evolution of Retrieval

From Ranked Lists to Generative Evidence

Traditional IR: Ranked lists optimized for precision, recall, and user clicks.

RAG: Needs evidence retrieval that supports generation, not navigation.

What matters now:

- Faithfulness, coverage, reasoning chains
- Retrieval that supports answer synthesis, not just result finding

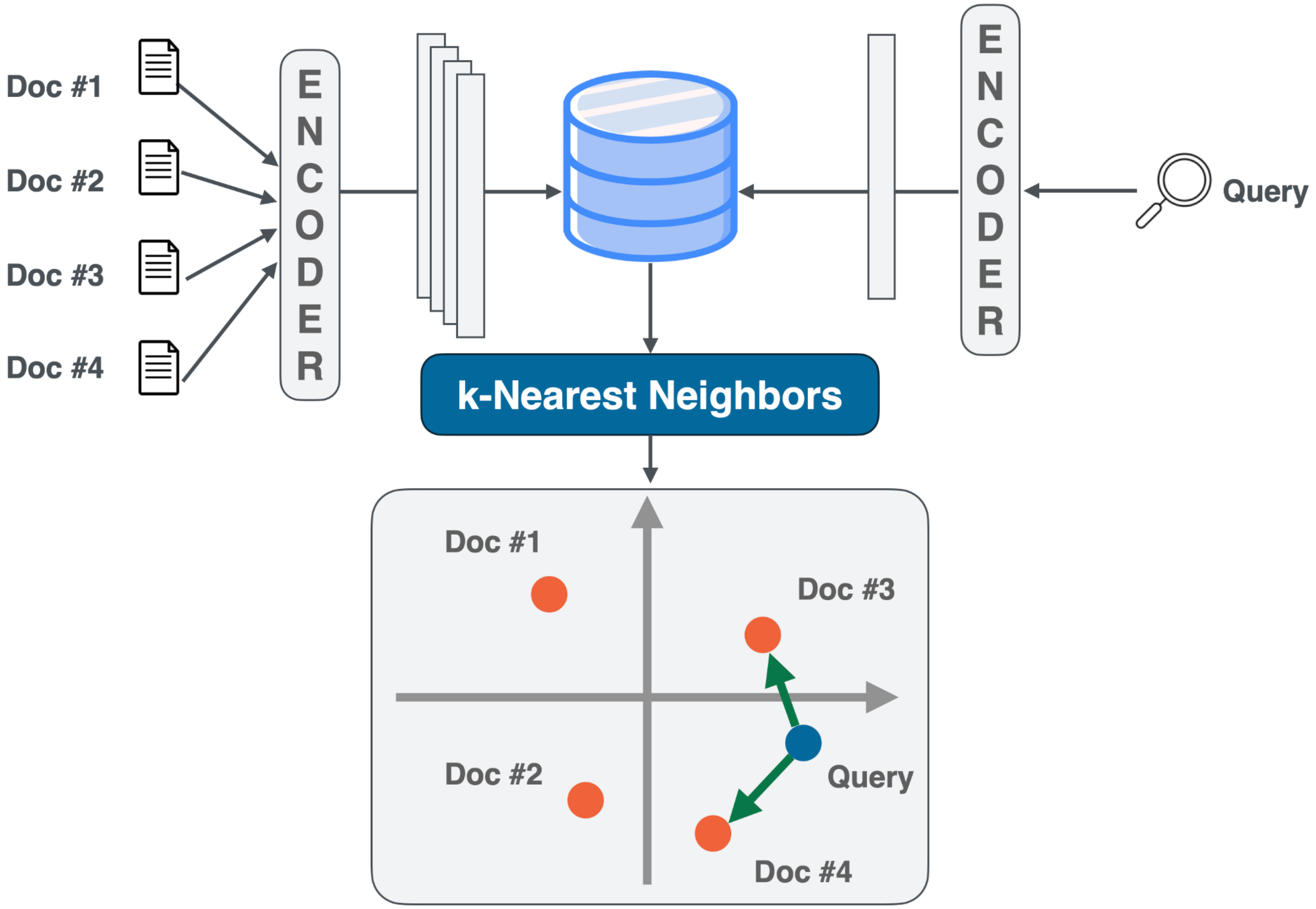
RAG Breaks the Old Rules

In RAG, the retriever isn't helping a user — it's helping a model. That changes everything.

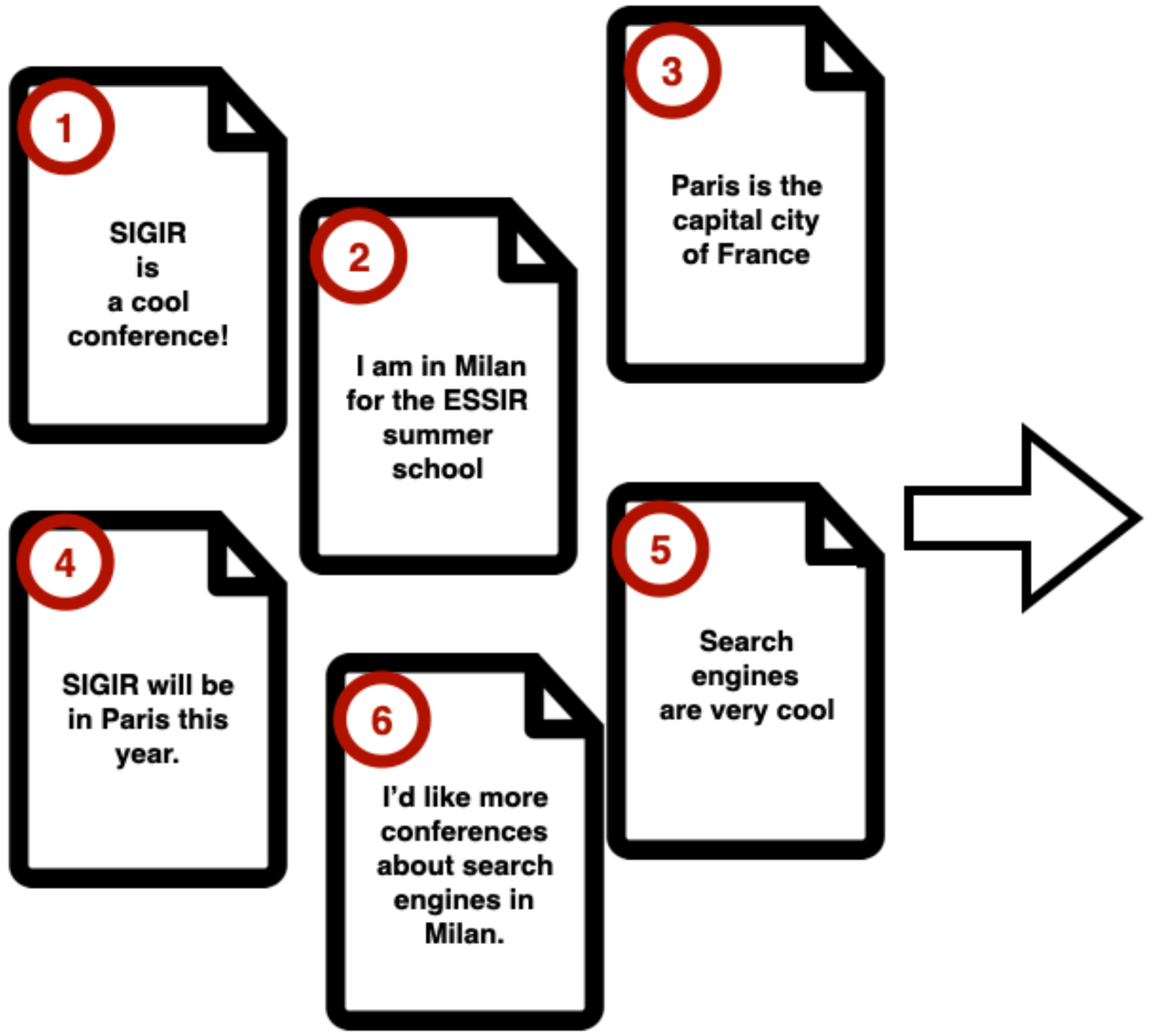
Traditional IR	RAG Systems
User browses ranked list	LLM consumes retrieved content
Optimize for top-k relevance (MAP, NDCG)	Optimize for grounding and synthesis
Queries are short, keyword-based	Queries are natural, long-form, ambiguous
Focus on document ranking	Focus on passage evidence selection
Evaluation: clicks, rank positions	Evaluation: faithfulness, hallucination rate
Retrieval ends with the user	Retrieval fuels generation

First Stage Retrieval

Dense Retrieval



Sparse Retrieval



T_1	capital	→	[3]
T_2	city	→	[3]
T_3	conference	→	[1, 6]
T_4	cool	→	[1, 5]
T_5	engine	→	[5, 6]
T_6	essir	→	[2]
T_7	france	→	[3]
T_8	milan	→	[2, 6]
T_9	paris	→	[3, 4]
T_{10}	school	→	[2]
T_{11}	search	→	[5, 6]
T_{12}	sigir	→	[1, 4]
T_{13}	summer	→	[2]
T_{14}	year	→	[4]

Terms = {capital, city, conference, cool, engine, essir, france, milan, paris, school, search, sigir, summer, year}

Sparse Retrieval Ubiquitous at LiveRAG

Nearly all* LiveRAG teams used some form of sparse in their pipeline

Most teams used BM25

CIIR (used Lion)

DoTA-RAG (prune with BM25)

GraphRAG used BM25 to seed a graph

Ragmatazz indexed with two BM25

* 17 out of 19

Sparse Retrieval @ LiveRAG

Ragmatazz

Rank	bm25s	bm25s_hyde	arc-embed	arc-embed_hyde
1	**.216**	.137	.15	.133
3	**.341**	.236	.267	.231
5	**.396**	.284	.337	.28
10	**.472**	.365	.424	.364
20	**.546**	.458	.518	.454
40	**.616**	.532	.615	.541
100	.71	.635	**.713**	.651
1000	.877	.844	**.882**	.853

Table 2: Mean Recall @ Rank for the 4 Retrievers

RAGentA

Table 2: Evaluation of Retrieval Performance

Retrieval System	MRR@20	Recall@20
BM25	0.4205	0.5020
E5	0.3476	0.4920
Hybrid	**0.4290**	**0.5650**

PreQRAG

Single-doc Questions				
	Top1	Top2	Top3	Top 10
Sparse (Rewritten)	37.5%	47.7%	52.1%	67.4%
Dense (Rewritten)	31.2%	36.2%	42.0%	52.1%
Multi-doc Questions				
	Top1	Top2	Top3	Top 10
Sparse (Rewritten)	32%	36%	40%	55%
Dense (Rewritten)	26%	30%	36%	56%

Table 2: Retrieval Performance Metrics for Single-doc and Multi-doc Questions

TopClustRAG

TABLE I

RETRIEVAL PERFORMANCE OF SPARSE, DENSE, AND HYBRID SYSTEMS ON THE SYNTHETIC VALIDATION SET.

System	MRR	R@1	R@5	R@10	R@50	R@100	R@200	R@1000
Sparse	**0.3361**	**0.2037**	**0.4074**	**0.4815**	0.5741	0.6481	0.7593	0.8704
Dense	0.0526	0.0000	0.0926	0.1111	0.2963	0.3333	0.3519	0.5556
Hybrid	0.1322	0.0370	0.1111	0.3519	**0.6852**	**0.7778**	**0.8519**	**0.8889**

Team Marikarp

Retriever	R@10	R@100	R@200	R@400	R@1k	R@2k	R@4k
sparse&dense	0.61	0.81	0.86	0.89	0.93	0.94	0.95
dense	0.53	0.72	0.77	0.84	0.88	0.90	0.93
sparse	0.59	0.76	0.82	0.86	0.89	0.93	0.94

Table 2: Recall of gold documents under single-document dataset using different retrieval methods.

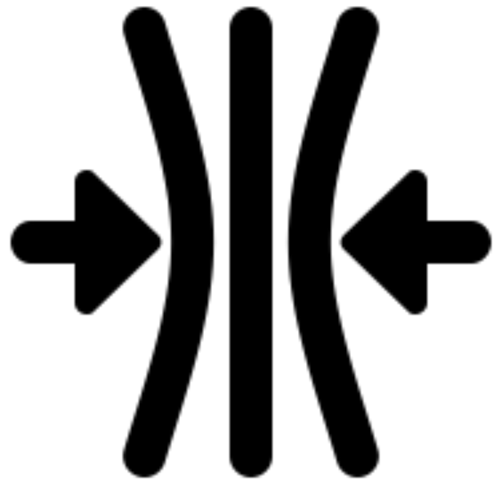
DoTA-RAG

Method	Correctness [-1:2]		Faithfulness [-1:1]	
	All words	300-word cap	All words	300-word cap
Baseline	0.752	0.761	-0.496	-0.493
+ Arctic-M	1.616	1.626	-0.216	-0.225
+ Routing	1.562	1.577	-0.108	-0.090
+ Pruning	1.562	1.566	0.428	0.404
+ Rerank	**1.652**	**1.686**	**0.672**	**0.662**
+ Rewrite	1.478	1.484	0.640	0.620

Emorag

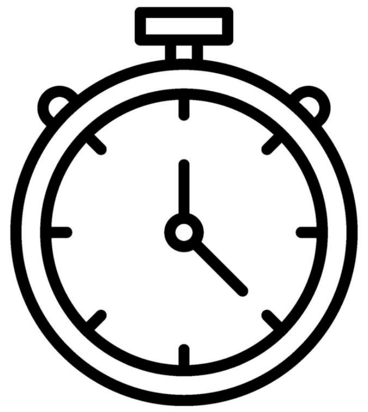
Name	MAP	Recip. Rank	nDCG@10	Recall@1	Recall@10
Sparse (OpenSearch BM25)	.523	.347	.497	.285	.485
Dense (Pinecone E5)	.352	.260	.367	.190	.435
Hybrid	.523	.347	.497	.285	.485

Efficiency



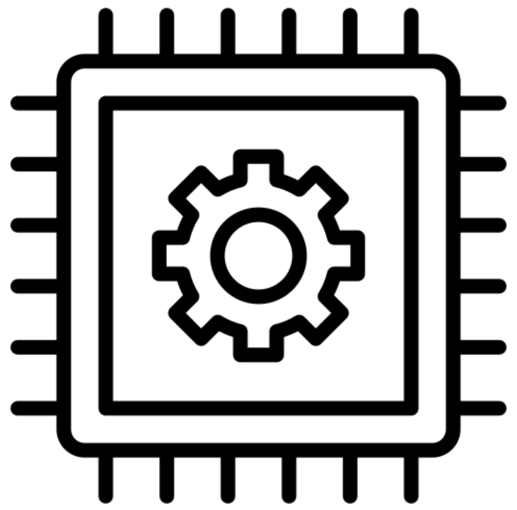
Memory footprint

Chen et al. (2022, EMNLP Findings) note that a BM25 inverted index for the MS MARCO passage corpus occupies only about 0.7 GB, whereas a dense retriever's vector index can require tens of gigabytes (approximately 26 GB in their example)



Latency and Query Throughput

Lin (2024, arXiv:2409.06464v1) shows that on the 15-million-document corpus a BM25 inverted index handles ~210 queries per second compared with ~56 QPS for a strong dense HNSW retriever, demonstrating that sparse retrieval can be faster than dense retrieval at large scale.



Compute Requirements and Deployment Cost

Lassance & Clinchant (2022) emphasize, “multi-CPU core + GPU (vs. mono-CPU core)” setups are often “the norm” for dense retrievers

Not a replacement for dense!

Sparse retrieval captures lexical overlap: precise, interpretable, efficient.

Dense retrieval captures semantic similarity: fuzzy matching, generalization.

Combined they consistently outperform either one alone.

A Proposed Conceptual Framework for a Representational Approach to Information Retrieval. Lin. SIGIR Forum 2021.

How to combine dense and sparse?

Score Fusion

Combine the scores returned by the sparse and dense retrievers.

RAGentA NoobRAG

Alternating merge

Alternately selecting from sparse or dense

Team Marikarp

Reciprocal Rank Fusion (RRF)

Instead of scores, use rank positions

RMIT–ADM+S TopClustRAG UiS-IAI Ragmatazz PRMAS-DRCA

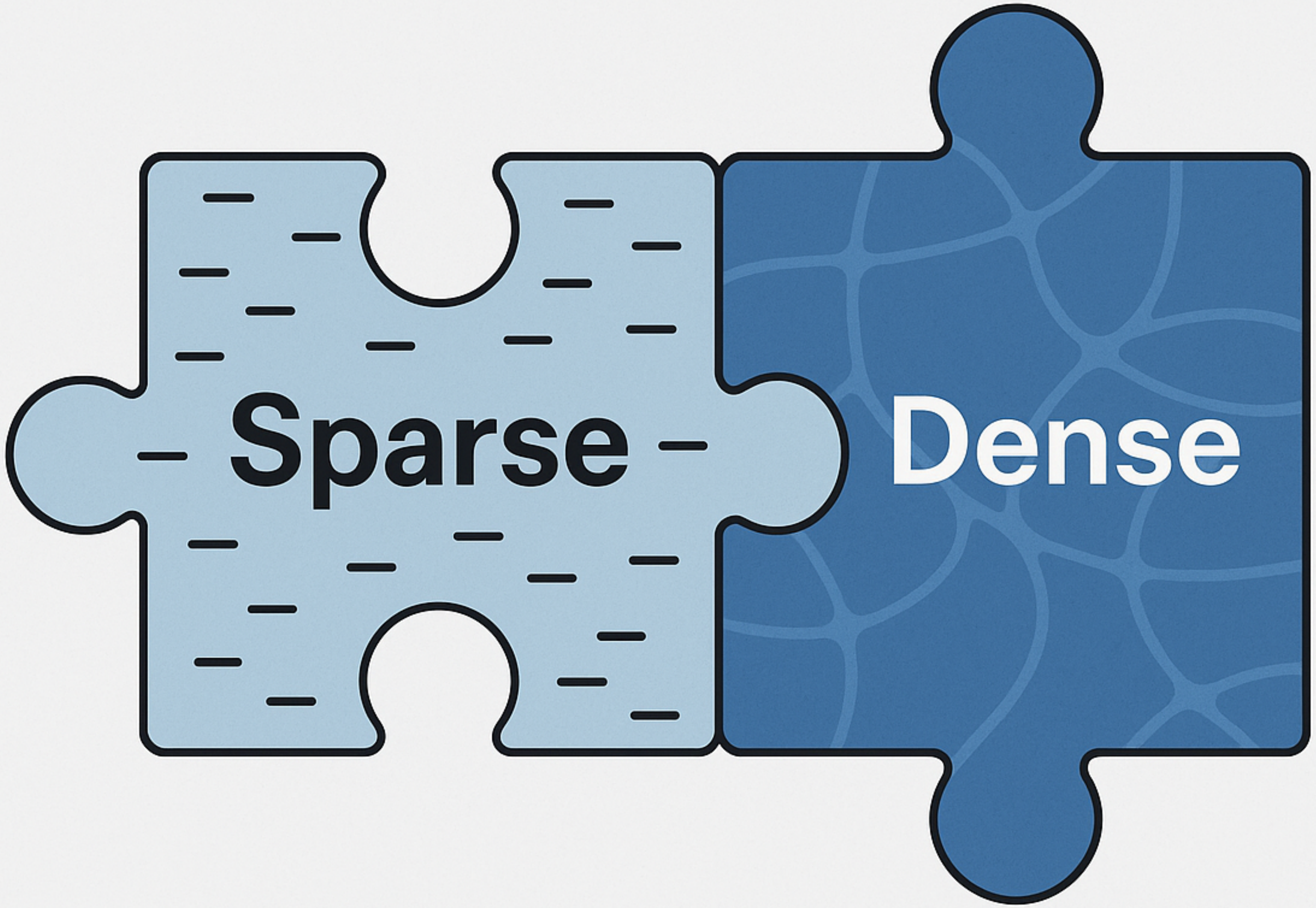
Retrieve + Rerank

Retrieve w/ sparse and dense ⟶ then use a reranker to find common ranking

Retrieve + Prune

Retrieve w/ dense ⟶ Sparse-based Pruning

DoTA-RAG



Hybrid and Sparse Retrieval Drive Better Coverage

"Disabling BM25 reduces Recall@5 by thirteen points because dense retrieval alone still struggles with misspellings and rare entities" - **TinyUPR**

Table 2: Evaluation of Retrieval Performance

Retrieval System	MRR@20	Recall@20
BM25	0.4205	0.5020
E5	0.3476	0.4920
Hybrid	**0.4290**	**0.5650**

"Our hybrid retrieval system achieves an MRR@20 of 0.4290, outperforming BM25 (0.4205) by +2.0% and E5 (0.3476) by +23.4%. " - **RAGentA**

Complementarity Objectives for Hybrid Retrieval

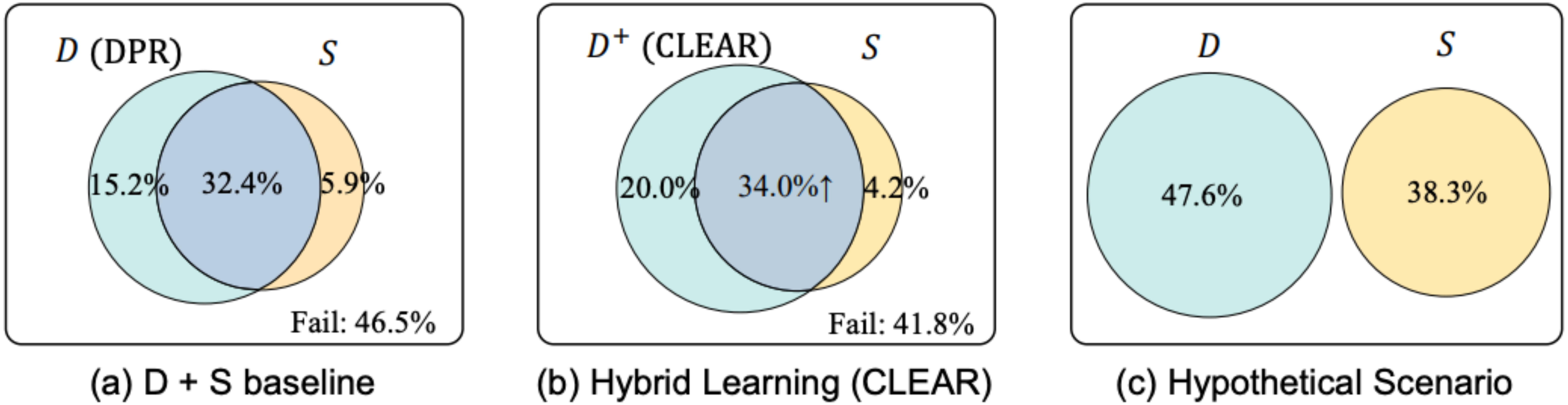


Figure 1: Recall@10 on Natural Questions. In the venn diargram, (a) shows BM25+DPR baseline and (b) shows CLEAR using residual margin. (c) is a hypothetical scenario, identical to (a) but without the intersection

On complementarity objectives for hybrid retrieval. Lee et al. ACL 2023.

What most teams use vs. what's available

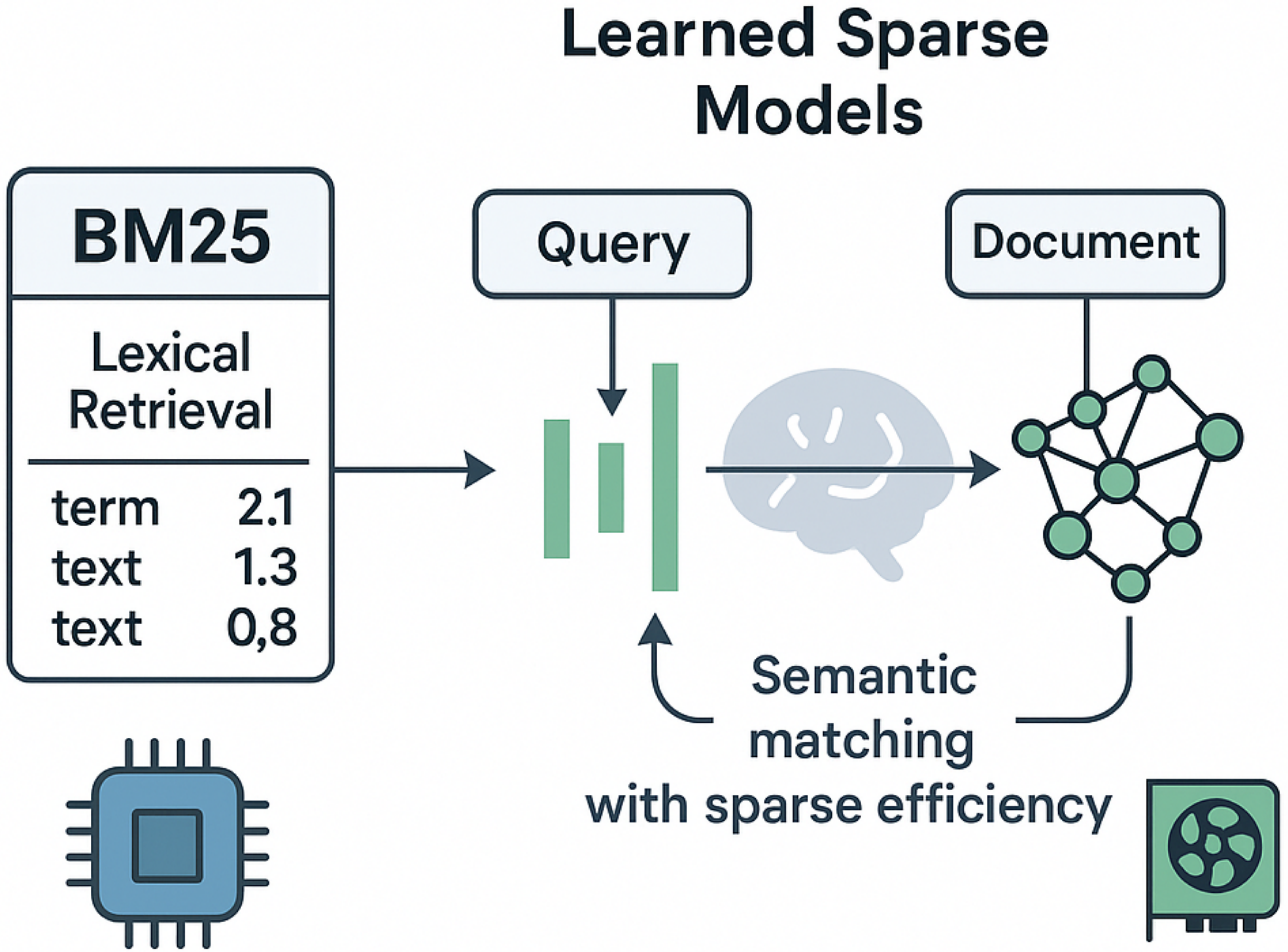
Despite sparse retrieval being everywhere, few teams moved beyond BM25.

It's a decades-old method, fast and solid, but doesn't reflect recent progress in sparse neural IR.

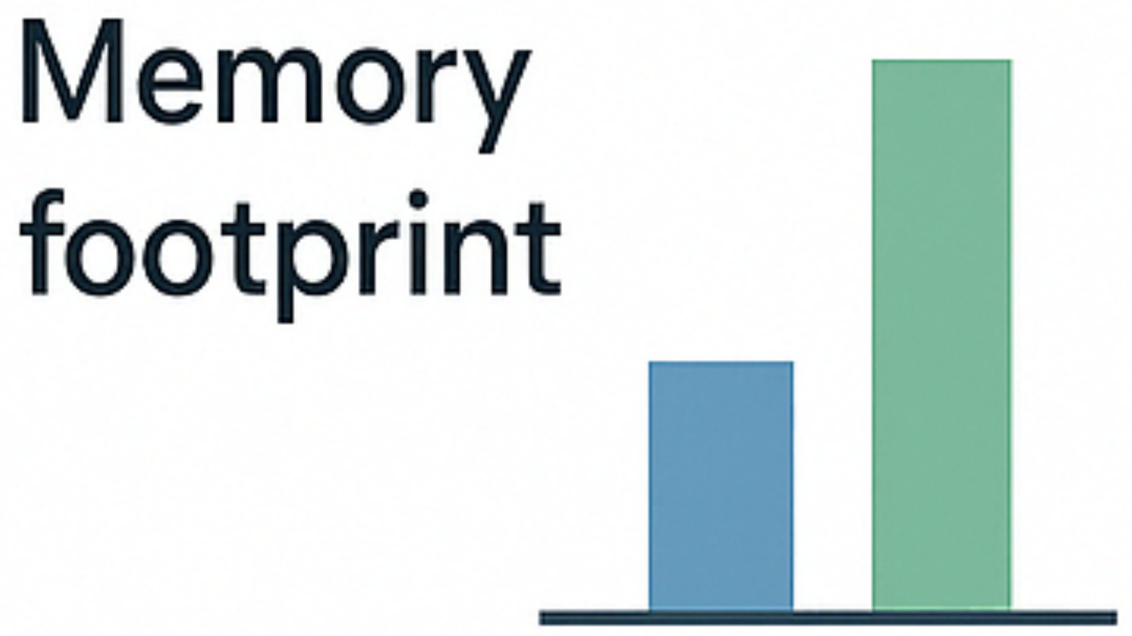
History of learned sparse models

* Not an exhaustive list

Enter Learned Sparse Models



- Semantic matching with sparse efficiency
- Term importance tailored to task
- End-to-end trainaibility for modern pipelines



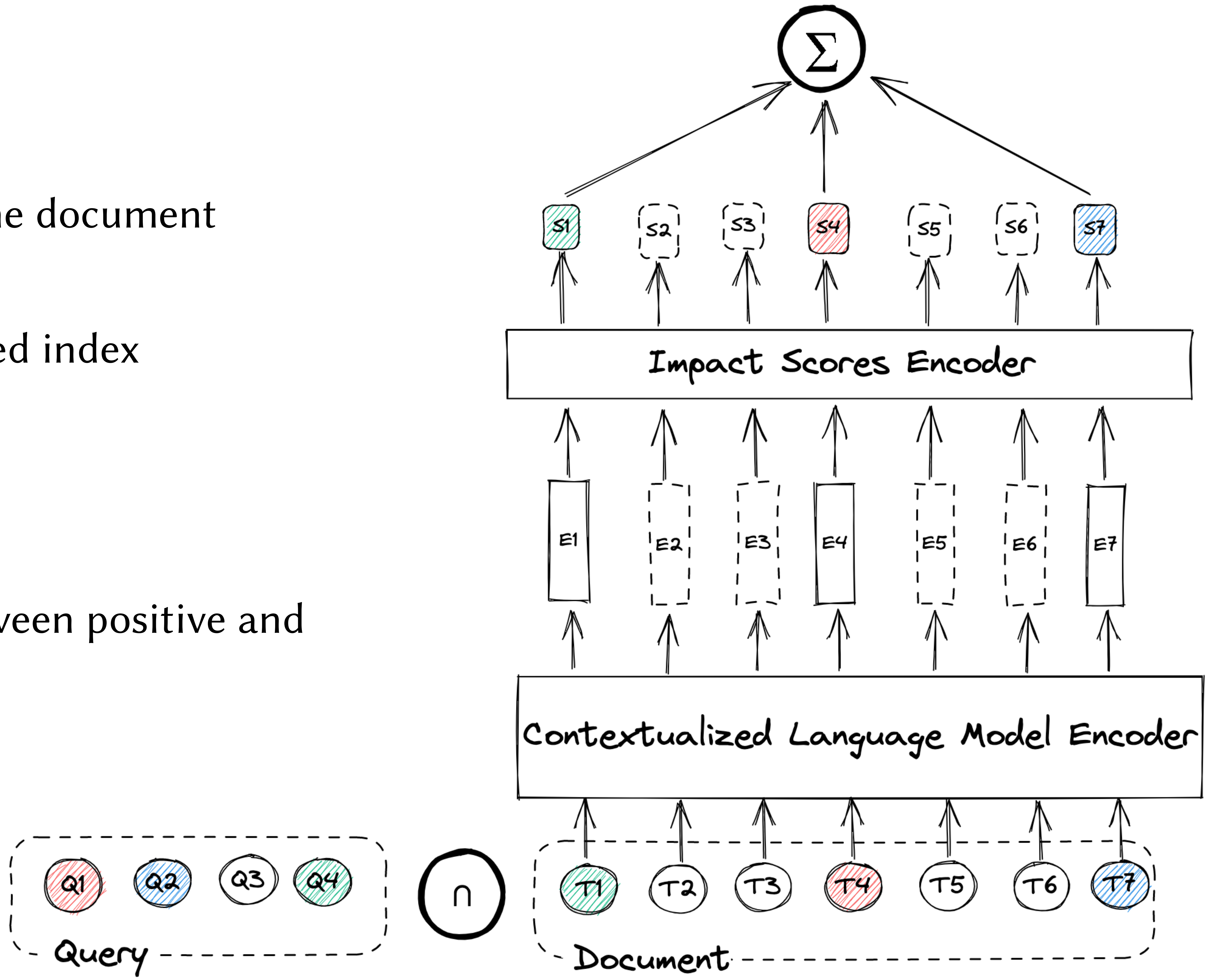
An example

Predict scores for each unique term of the document

Directly store quantized scores in inverted index

Score is the sum of intersection terms

Goal: maximize the score difference between positive and negative document



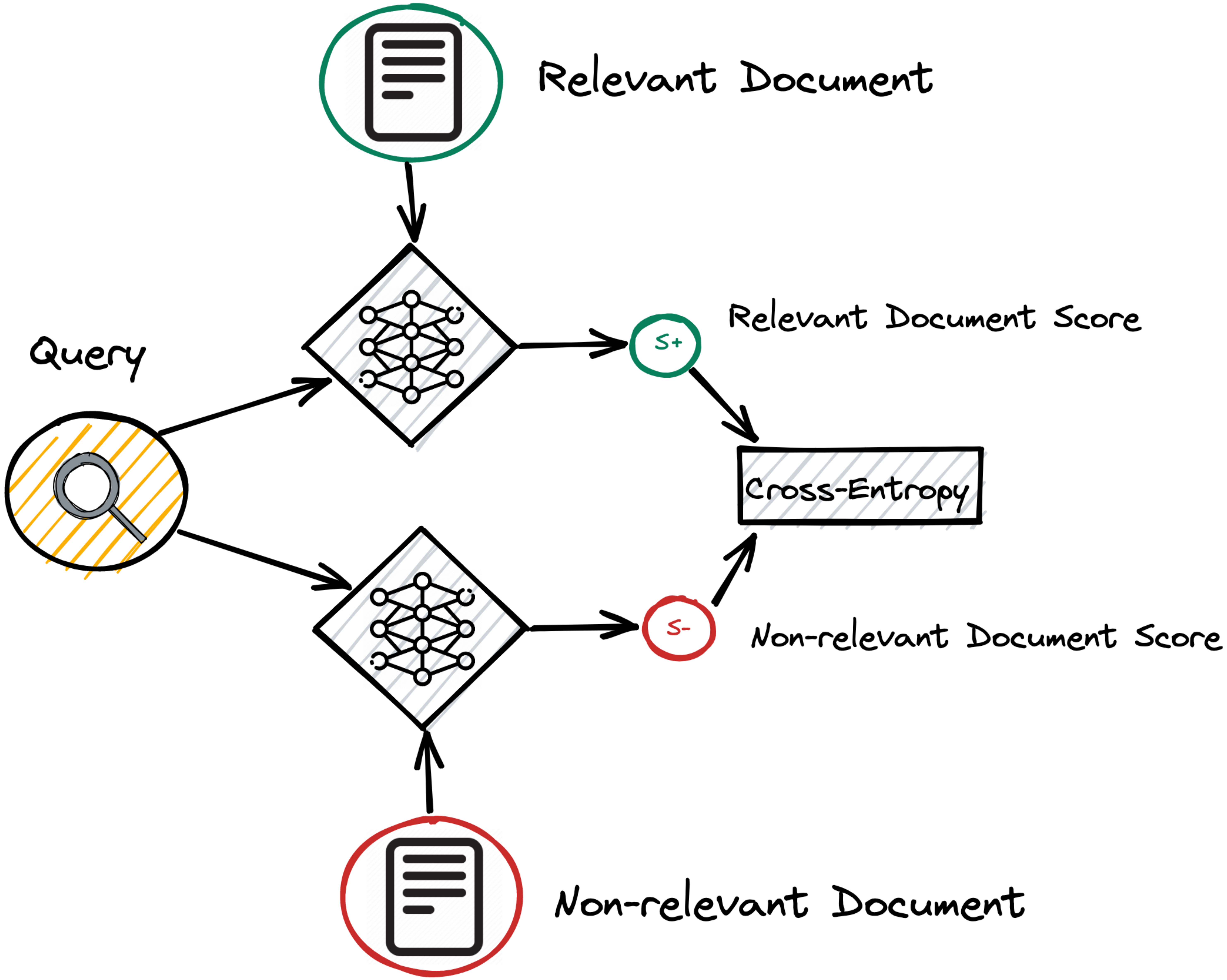
Training a sparse model

Triples sampled from the MS-MARCO training dataset

A query, a relevant passage, and a presumed non-relevant passage per sample

Two scores for the corresponding two documents are computed

Pairwise softmax cross-entropy loss over the document scores

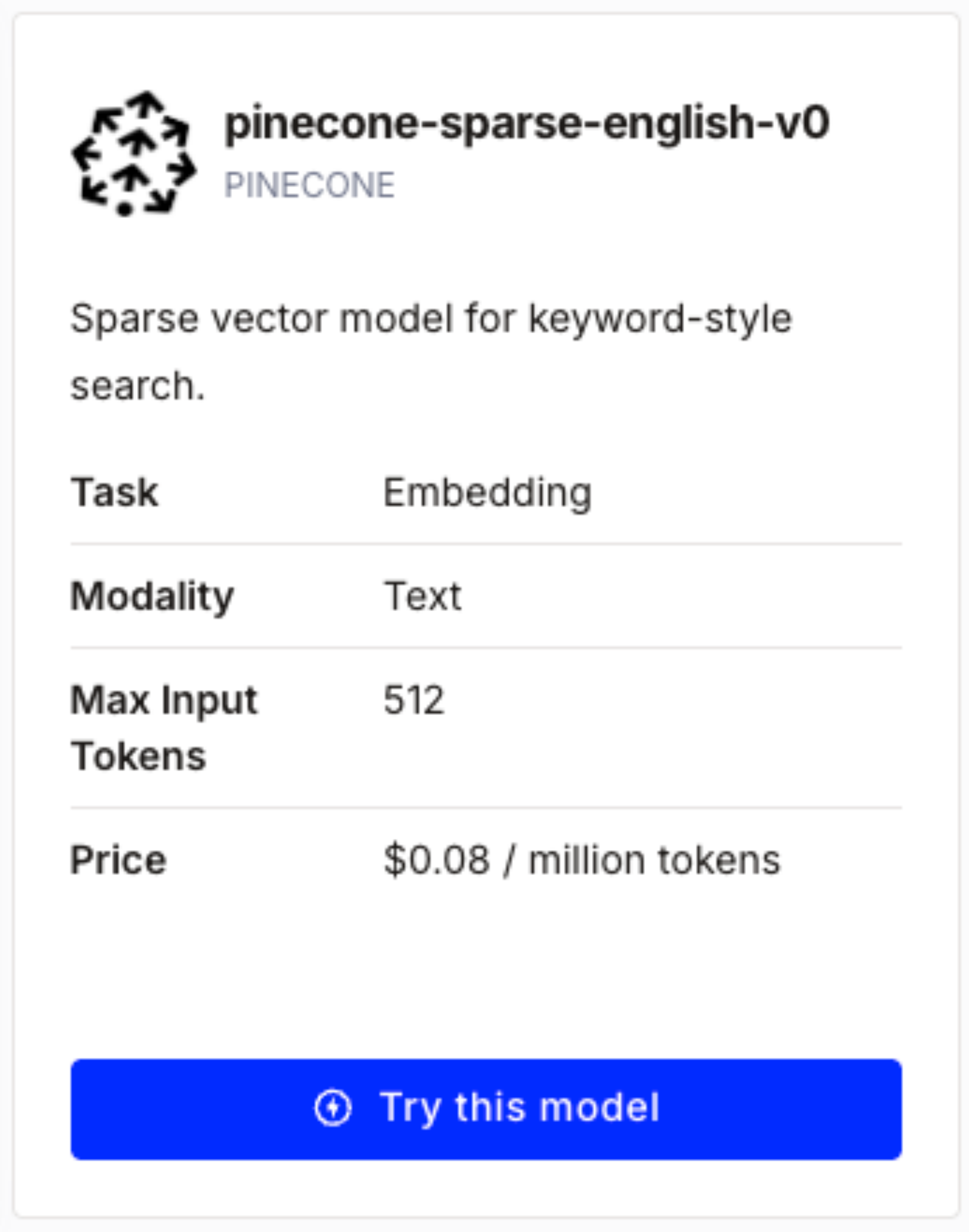


Pinecone Sparse Model

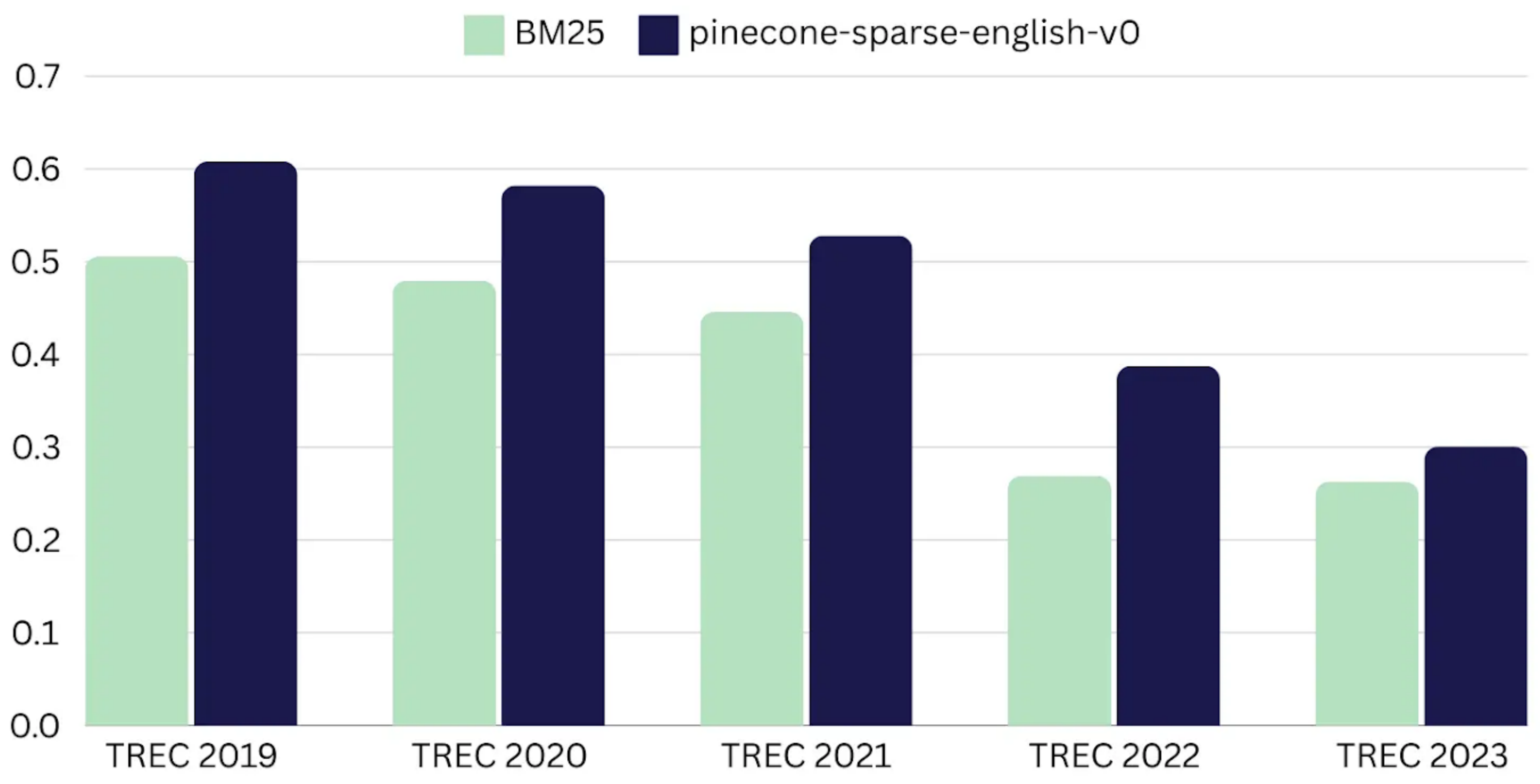
Built on top of the DeepImpact architecture

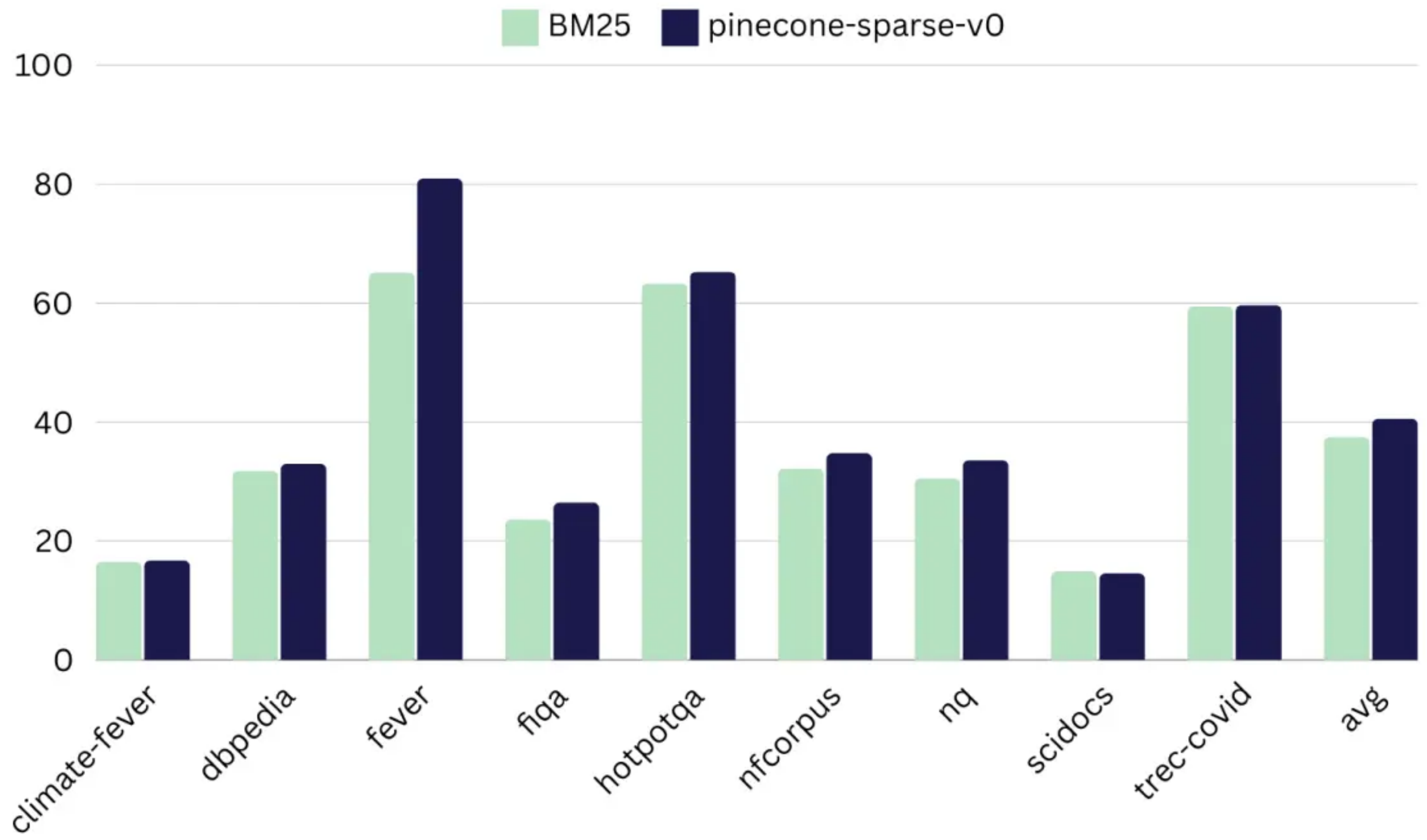
The model directly estimates the lexical importance of tokens by leveraging their context

Outperforms BM25 by up to 44% (average 23%) NDCG@10 on Text Retrieval Conference (TREC) Deep Learning Tracks and up to 24% (8% on average) on BEIR.



Benchmarking pinecone-sparse-english-v0





Contextual Importance

It leverages a combination of a **traditional inverted indexes** and **contextualized language models** for efficient retrieval

It estimates the semantic importance to produces a single value impact score for each tokens of a document collection

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?

Whole-word tokenization

"Top cardiologists in Palo Alto for mitral valve repair"

WordPiece tokenization


```
{
  "mit": 1.73,
  "alto": 1.7,
  "palo": 1.63,
  "valve": 1.52,
  "repair": 1.29,
  "hospital": 1.15,
  "##ral": 1.03,
  "heart": 1.03,
  "doctor": 1.02,
  "##ogist": 1.02,
  "##logist": 0.92,
  "top": 0.92,
  "clinic": 0.87,
  "surgery": 0.76,
  "card": 0.74,
  "surgeon": 0.68,
  "valves": 0.67,
  "dentist": 0.59,
  "##iol": 0.59,
  "clara": 0.53,
  "specialist": 0.52
}
```


Not limited to predefined tokens

(e.g. ~32k BERT vocabulary)

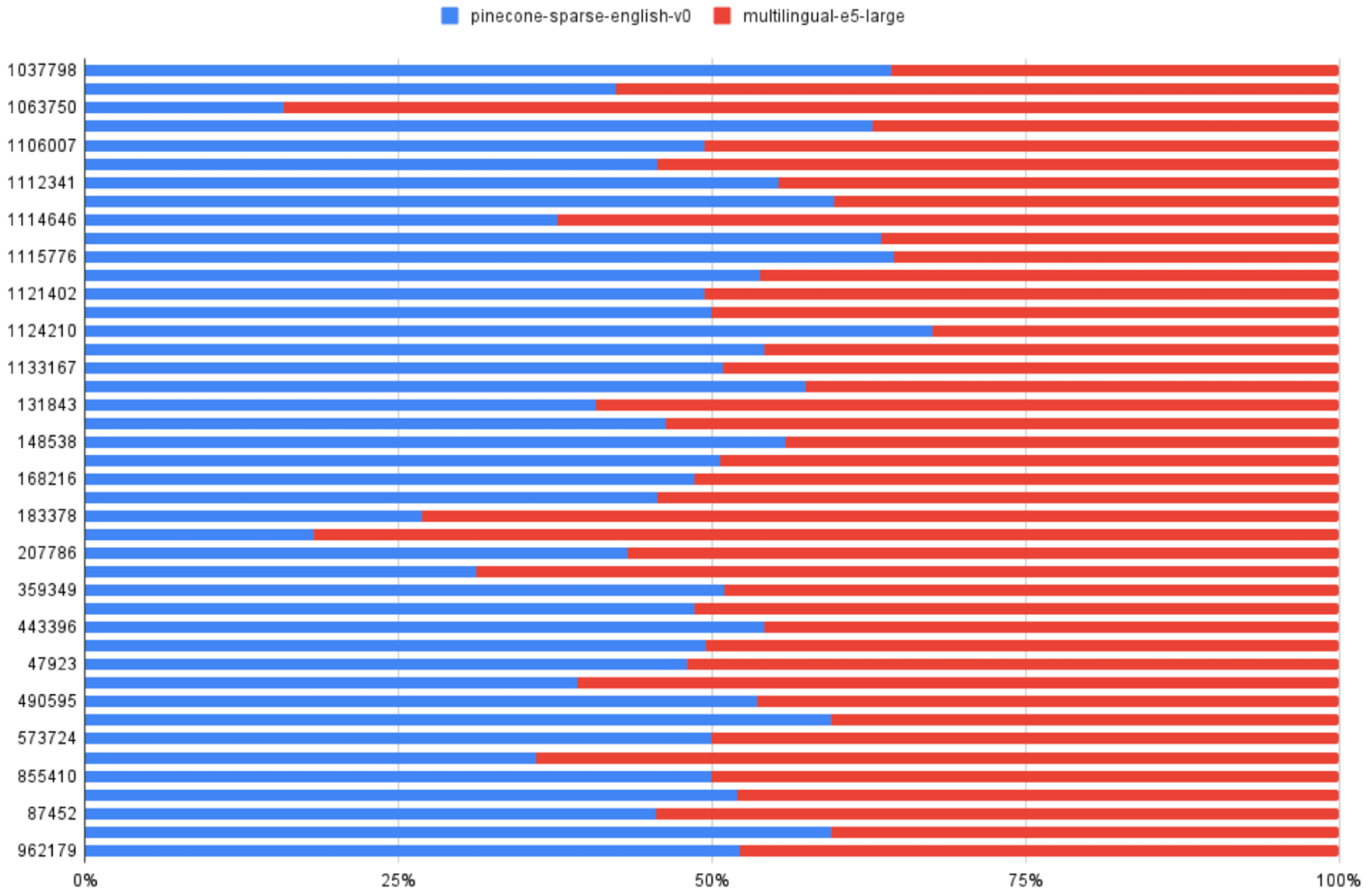
No undesired fragmentation of keywords

No addition of unrelated/harmful tokens

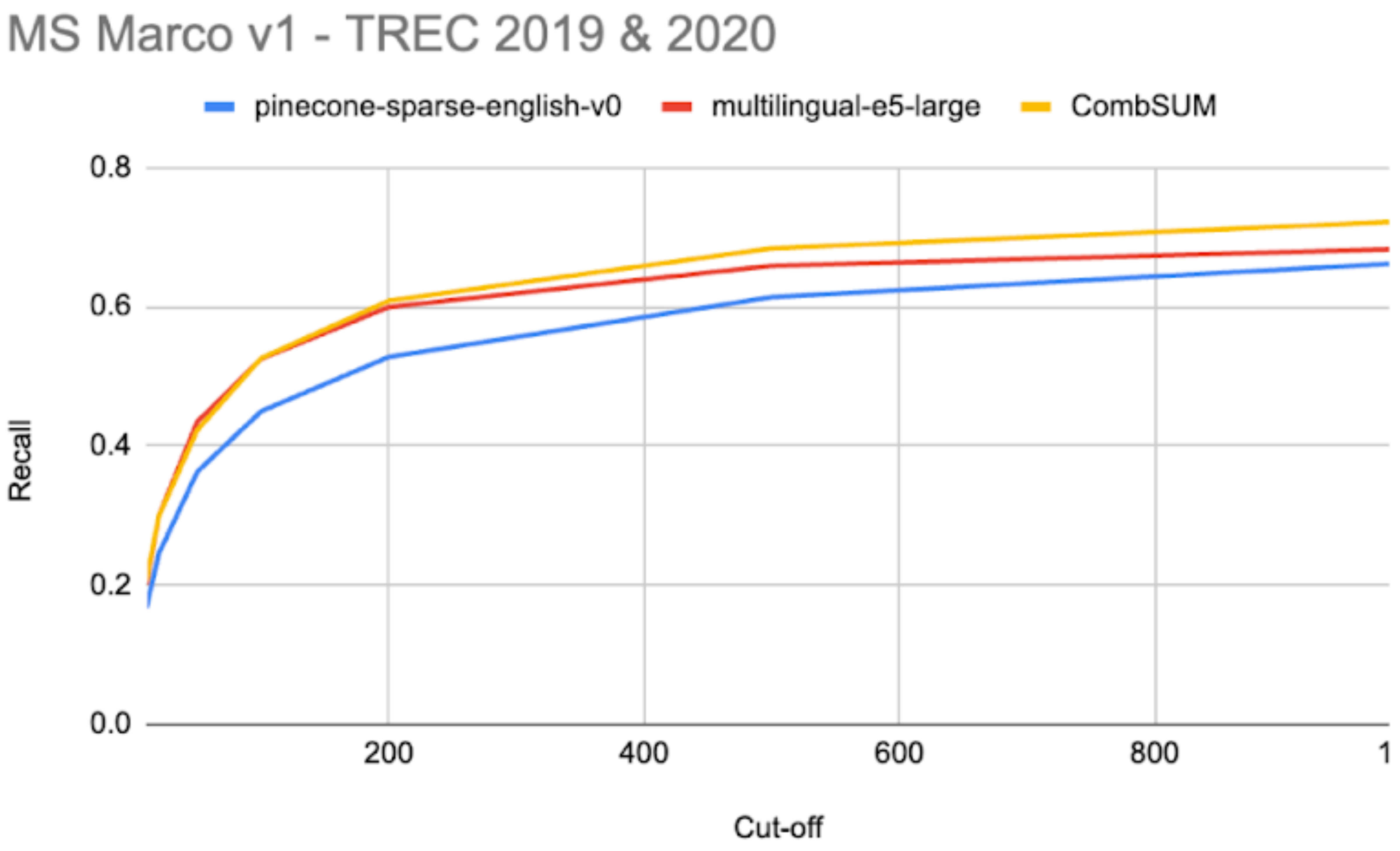
Whole-word tokenization

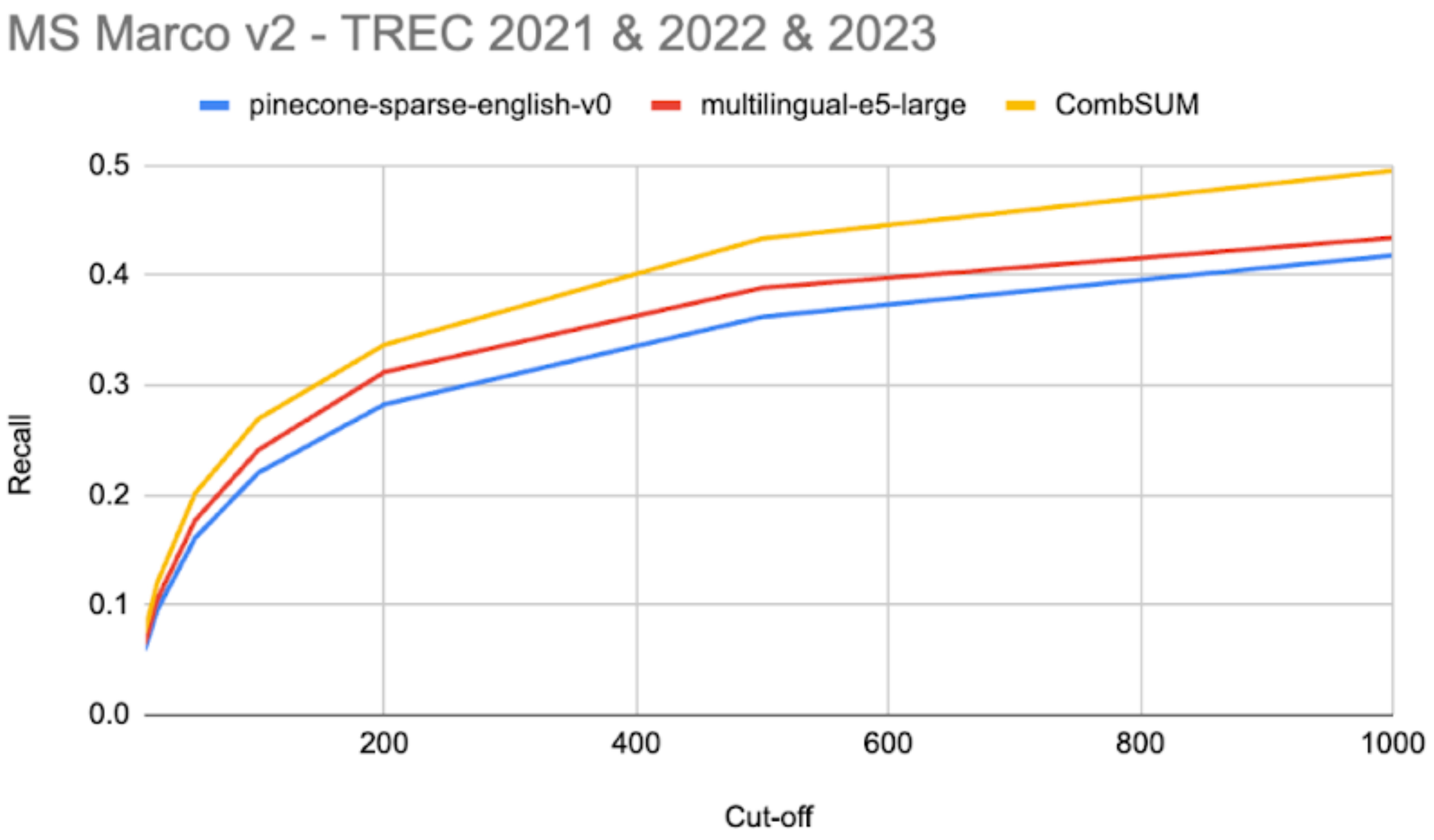

```
{
  "mitral": 4.95,
  "cardiologists": 4.55,
  "palo": 4.24,
  "alto": 3.78,
  "valve": 3.29,
  "repair": 2.31,
  "top": 1.33,
  "in": 0.8,
  "for": 0.76
}
```

Query-specific Retrieval Quality

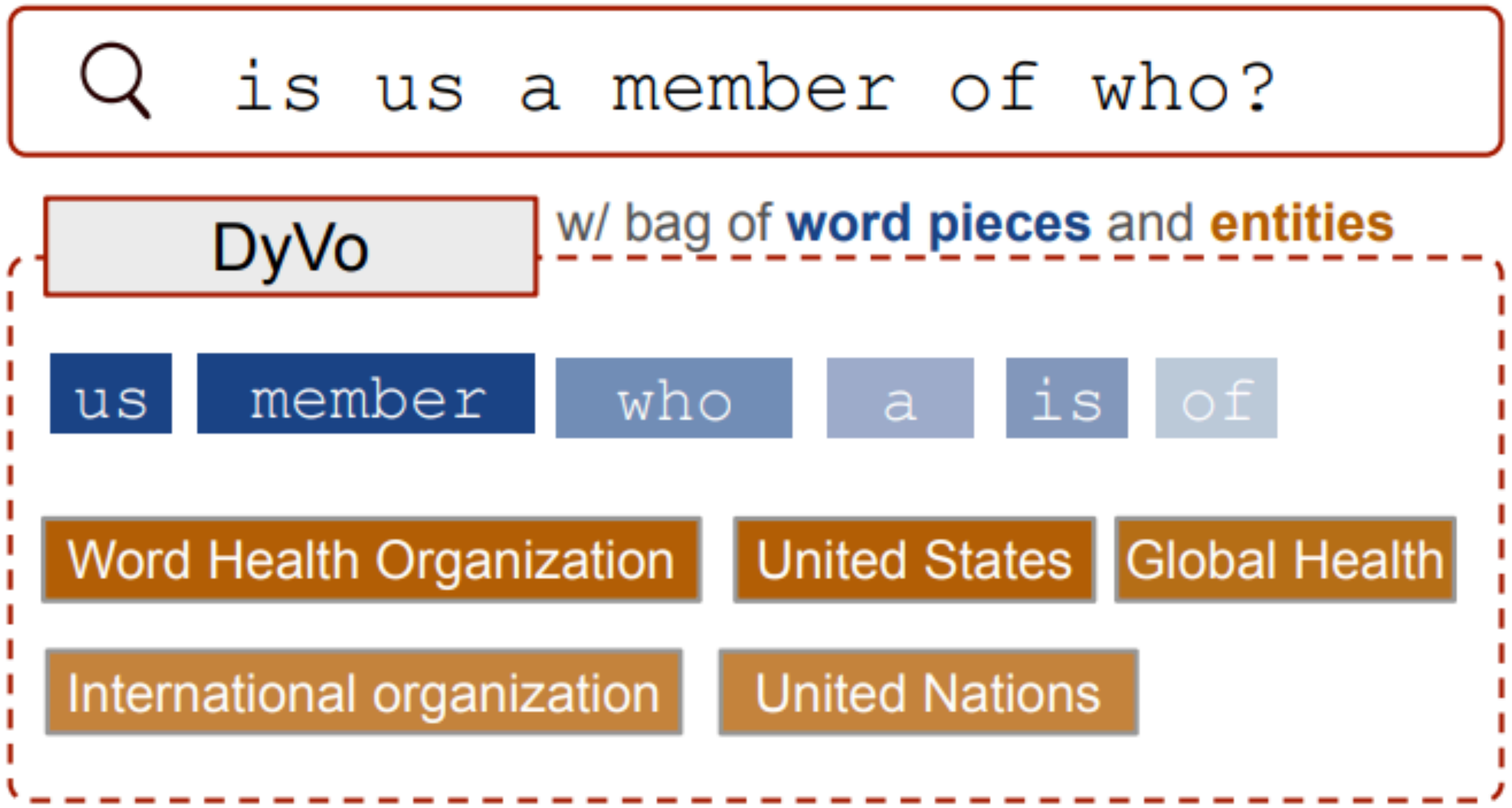


Combining Sources Improves Recall





Learned Sparse Retrieval with Entities



Method	Reg	TREC Robust04			TREC Core 2018			CODEC		
		nDCG@10	nDCG@20	R@1k	nDCG@10	nDCG@20	R@1k	nDCG@10	nDCG@20	R@1k
Unsupervised sparse retrieval										
BM25		39.71	36.25	57.18	30.94	29.19	52.19	37.70	35.28	61.25
BM25 + RM3		43.77	40.64	64.21	35.82	34.79	60.09	39.93	39.96	65.70
Zero-shot Dense Retrieval										
DistilBERT-dot-v5		37.95	34.97	52.41	37.02	34.60	54.07	42.76	46.67	60.33
GTR-T5-base		43.79	39.33	54.35	38.81	36.51	57.62	48.42	54.01	66.96
Sentence-T5-base		44.06	39.60	57.64	43.18	39.54	60.88	44.22	32.10	65.48
Learned Sparse Retrieval										
LSR-w	1e-3	40.37	37.23	55.66	34.50	31.45	52.66	39.10	35.32	57.58
DyVo (REL)	1e-3	**41.52**	**38.62**	**56.78**	**37.50**	**34.61**	**54.14**	**42.67**	**38.32**	**59.81**
LSR-w	1e-4	47.69	44.48	64.47	38.94	37.37	60.44	50.54	46.71	66.39
DyVo (REL)	1e-4	**48.15**	**44.85**	**64.72**	**43.10**	**39.46**	**60.43**	**51.66**	**47.95**	**68.49**
LSR-w	1e-5	49.13	46.34	66.86	40.99	38.73	63.22	52.61	49.22	69.07
DyVo (REL)	1e-5	**51.19**	**47.65**	**68.56**	**43.72**	**40.56**	**63.56**	**53.40**	**51.15**	**70.60**

Table 1: Results with linked entities. All LSR models use a DistilBERT backbone. DyVo uses entities found by the REL entity linker and LaQue entity embeddings. All documents are truncated to the first 512 tokens.

DyVo: Dynamic Vocabularies for Learned Sparse Retrieval with Entities. Nguyen et al. EMNLP 2024.

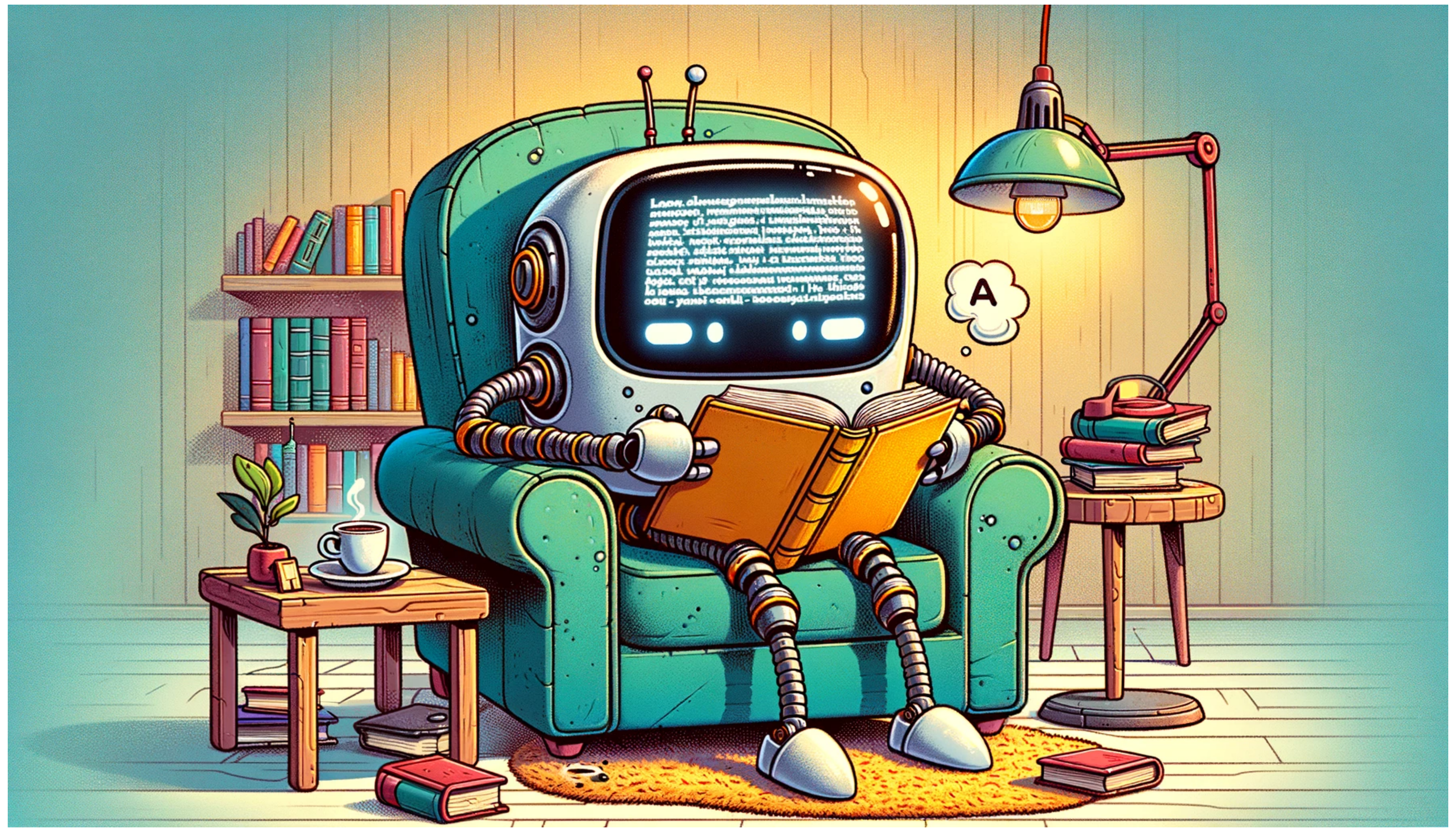
Rethinking Sparse Retrieval in the Age of RAG

✅ Treat sparse as a first-class citizen in RAG — not just a fallback for coverage.

🔬 Explore modern sparse models like DeepImpact, uniCOIL, and SPLADE that combine efficiency with semantic awareness.

🧩 Use sparse strategically for:

- 🔗 Evidence grounding (exact terms matter)
- 🔄 Multi-hop and graph-based reasoning
- ⚡ Low-latency, high-throughput deployments
- ⚖️ Evaluate sparse and dense under the same conditions — scale, tuning, budget.
- 💡 Design hybrid pipelines that do more than fuse — they collaborate.



“RAG may be a generation task — but it’s still retrieval-augmented. Let’s make that retrieval smarter, faster, and sparser.”

Thanks!

Any questions?