Beyond BM25: A Dense Retrieval Approach Using Sentence-BERT and FAISS

G7
Mohammed Al-Hashimi
Mustafa Al-Hashimi
Abubakar Passum Abdul Gaffar
Mehran Mamivand

Agenda

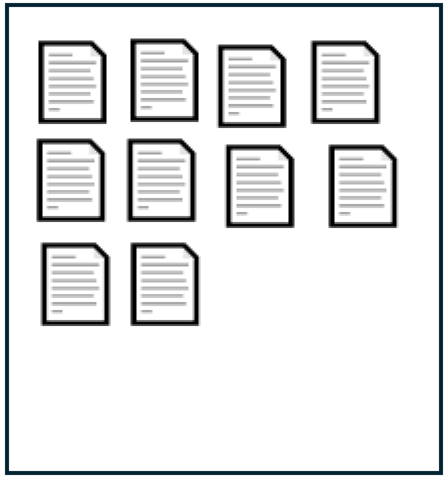
- Goal
- Dataset
- Related Work
- Method
- Results

Agenda

- Goal
- Dataset
- Related Work
- Method
- Results

Goal

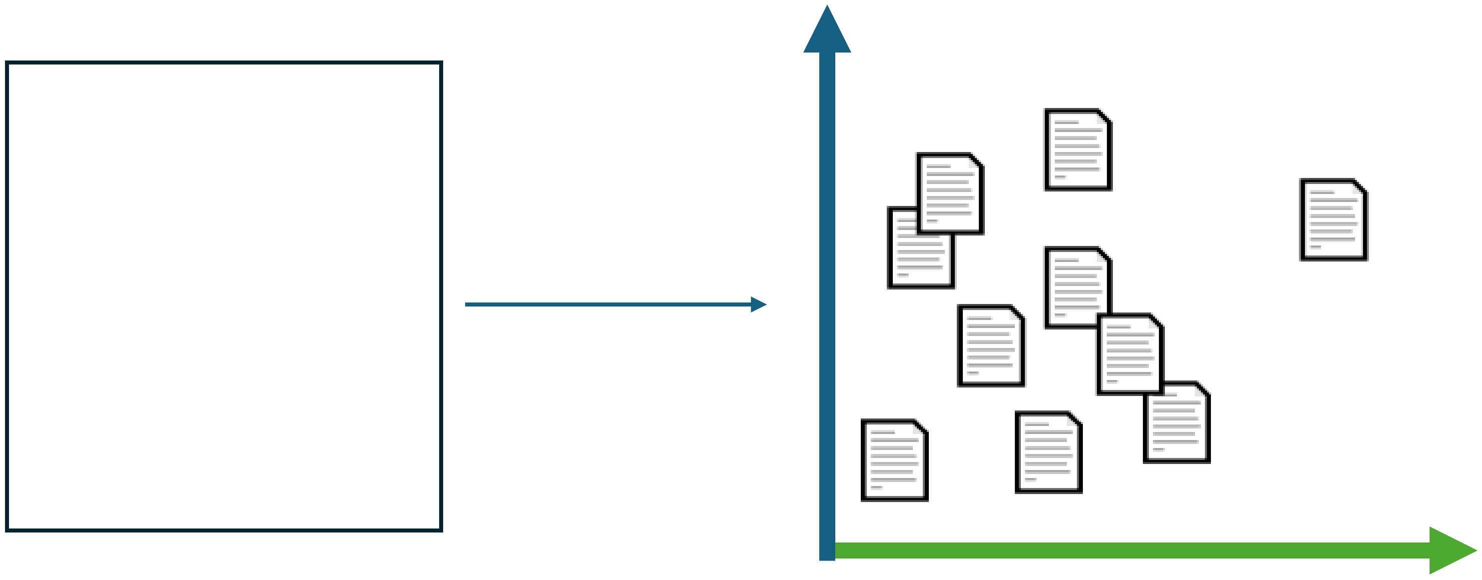
- **Given a query** → find the most relevant passages from a large collection



Collection / corpus

Goal

- **Given a query →** find the most relevant passages from a large collection

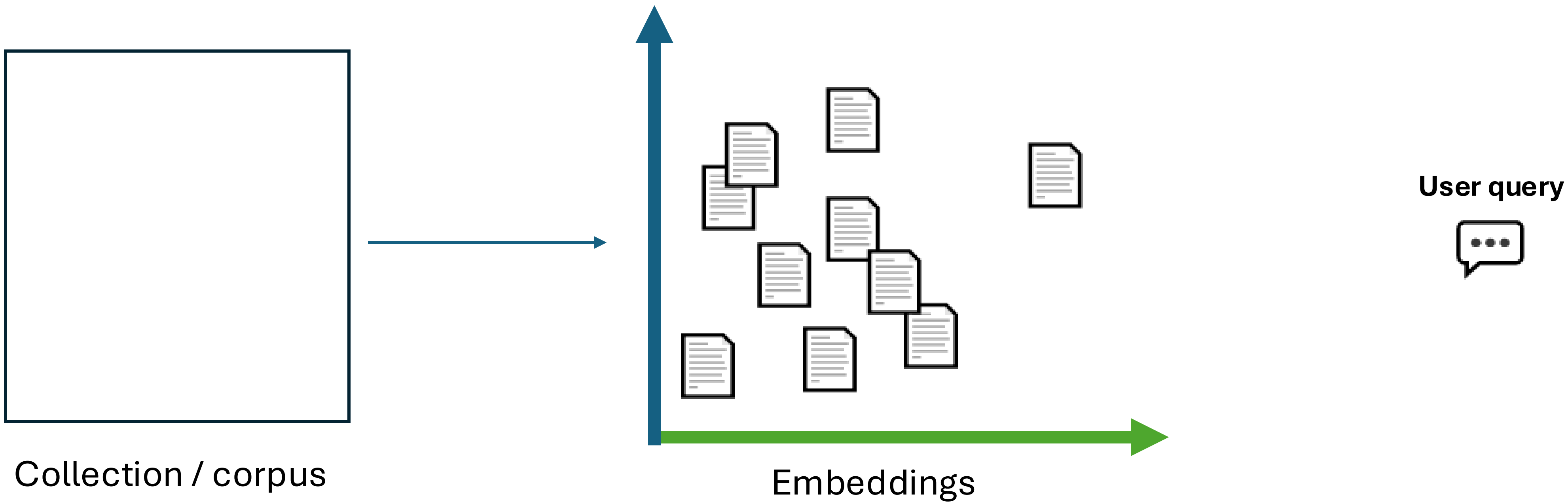


Collection / corpus

Embeddings

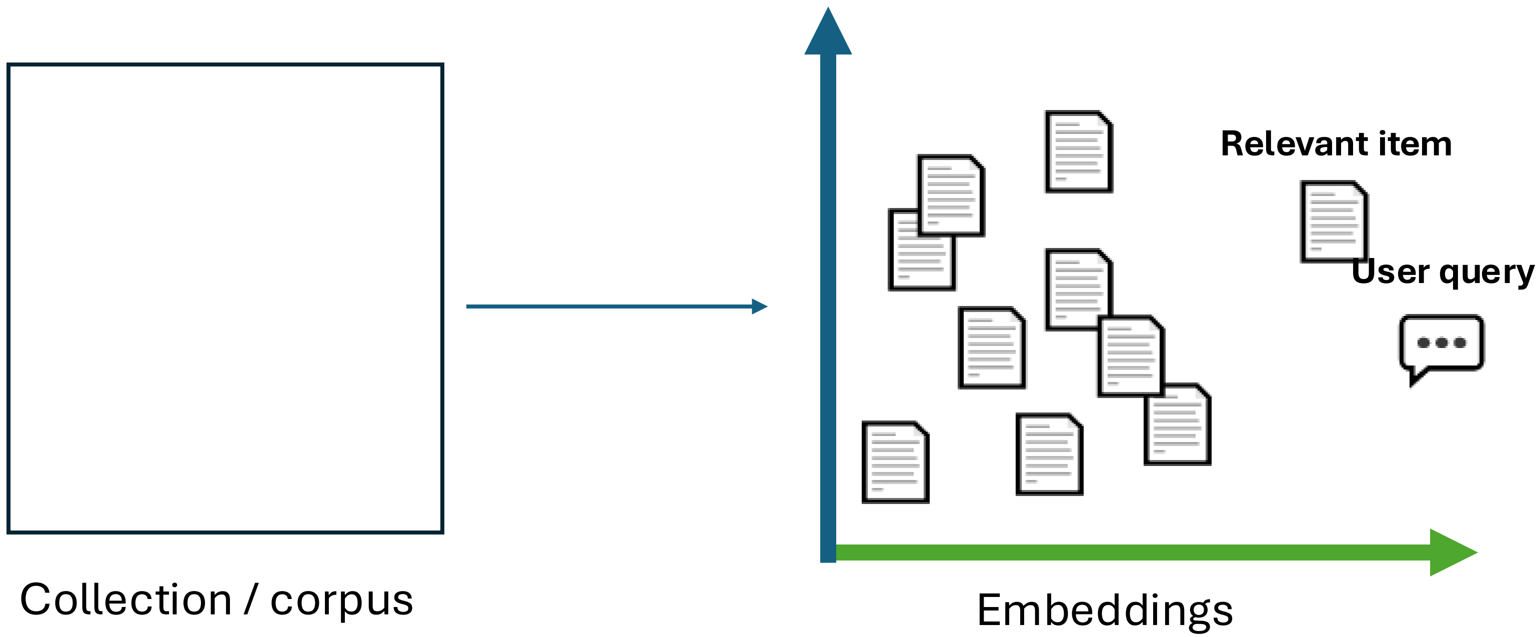
Goal

- **Given a query** → find the most relevant passages from a large collection



Goal

- **Given a query →** find the most relevant passages from a large collection



Agenda

- Goal
- Dataset
- Related Work
- Method
- Results

MS MARCO dataset

Microsoft **Ma**chine **R**eading **Co**mprehension

- **Passage ranking dataset** (retired in 2023)
- Collection of **8.8M passages** (pid, passage_text)
- **Queries 100K** (qid, query)
- **Qrels** (qid, pid) – human labeled relevance

MS MARCO Passage Ranking Leaderboard

Search:

	description	team	paper	code	type	date	eval	dev	t
🏆	**AliceMind Search LM (SLM) + Hybird List Aware Reranking (HLAR)**	Alibaba DAMO NLP Group & CTO Line-AI Engine Group	[paper]	[code]	full ranking	2022/03/17	0.450	0.463	
	Listwise + Fusion reranker	Liang Wang - MSR Asia			full ranking	2022/06/02	0.440	0.454	
🏆	**Anonymous**	Anonymous			full ranking	2022/02/16	0.439	0.455	
	Anonymous	Anonymous			full ranking	2022/03/02	0.439	0.453	
	CoT-MAE	Xing Wu (1), GuangYuan Ma(2) — Kwai NLP team (1), Knowledge Computing and Service Group, IIE, CAS (1,2)	[paper]	[code]	full ranking	2022/09/19	0.438	0.456	
🏆	**Lichee-xxlarge + deberta_v3-large + Reranking**	Lichee Team — Tencent QQBrowser NLP			full ranking	2021/12/10	0.436	0.452	
	Anonymous	Anonymous			full ranking	2022/01/12	0.435	0.450	

Link to full leaderboard

Agenda

- Goal
- Dataset
- Related Work
- Method
- Results

Related work

- R. Nogueira and K. Cho, **‘Passage Re-ranking with BERT’**, 2020.
  - Simple **two stage** method
  - <u>First stage</u>: Use **BM25** to pair queries and passages
    - **BM25** is a ranking function for text retrieval using TF-IDF (similar to BoW)
  - <u>Second stage</u>: Use **BERT**
  - <u>Results</u>
    - **BM25** MRR@10 = 16.7
    - **BM25 + BERT base** MRR@10 = 34.7
    - **BM25 + BERT large** MRR@10 = 36.5

- Other projects use advanced **three-stage** methods
  - Example results: MRR@10 = 39.7

Research questions

- **RQ1:** How does a **fine-tuned** embedding model perform against a **general-purpose** model on passage ranking?

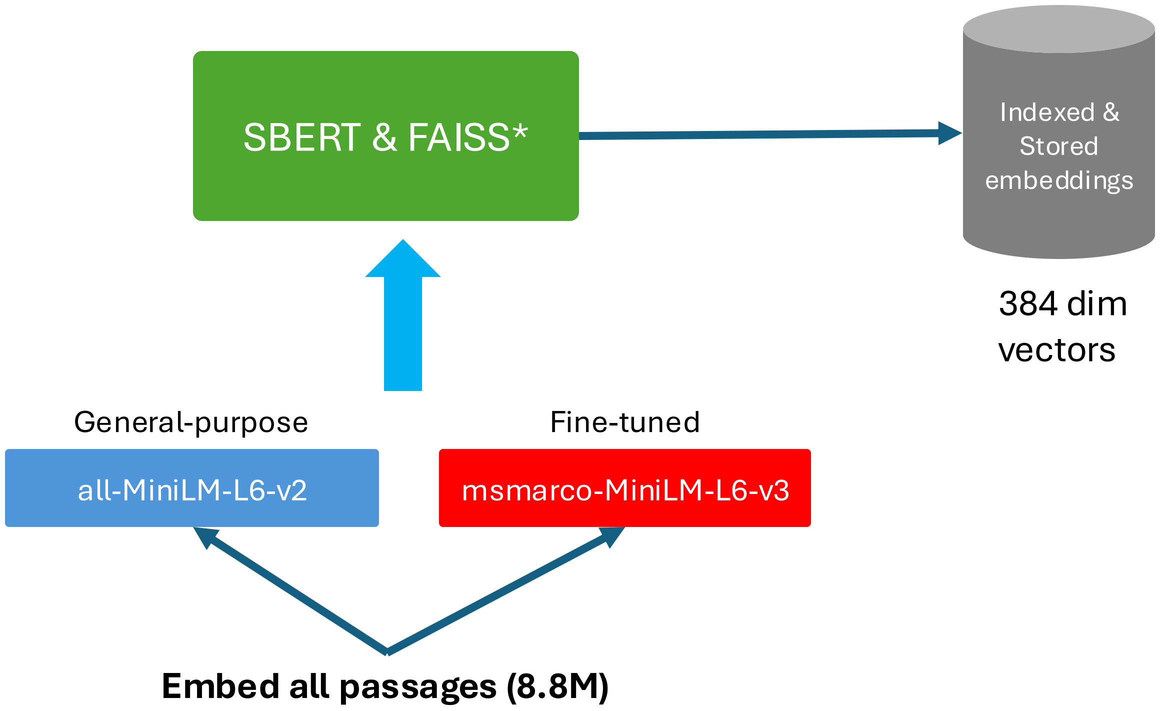
- **RQ2:** How do the results **compare** to other systems (including sparse retrieval BM25) found in the **MS MARCO leaderboard**?

Agenda

- Goal
- Dataset
- Related Work
- Method
- Results

Method

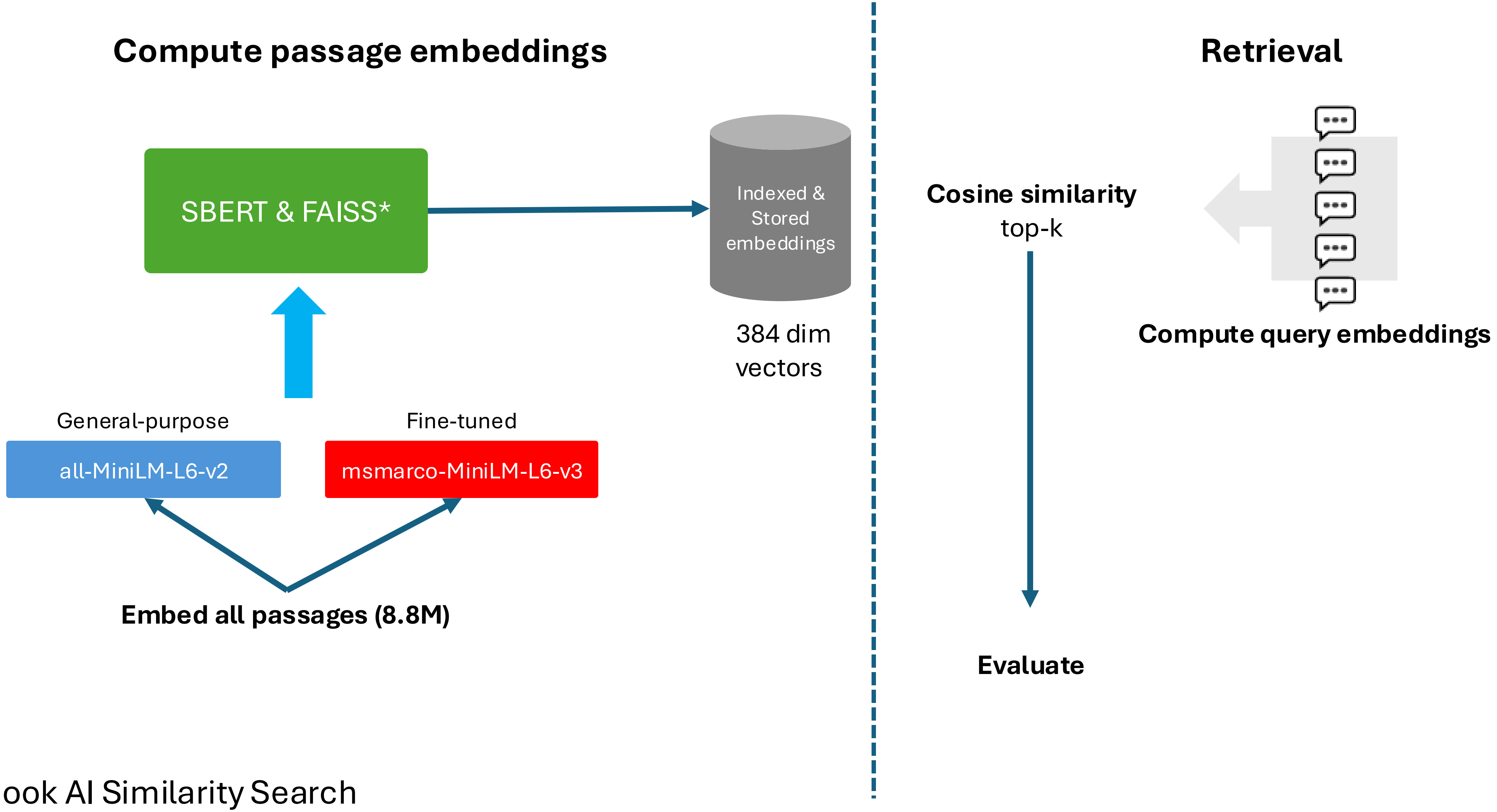
Compute passage embeddings



Retrieval

* Facebook AI Similarity Search

Method



* Facebook AI Similarity Search

Method

Compute passage embeddings

Entire collection **8.8M** rows (3 GB)

Due to hardware limitations, loading the entire collection into the RAM and converting to embeddings would result in slower compute. However, the real limit is GPU VRAM

float32 = 32 bits = 4 bytes
Embedding vector dimension = 384
For all passages: 8.8M * 384 * 4 ≈ **13.5 GB**

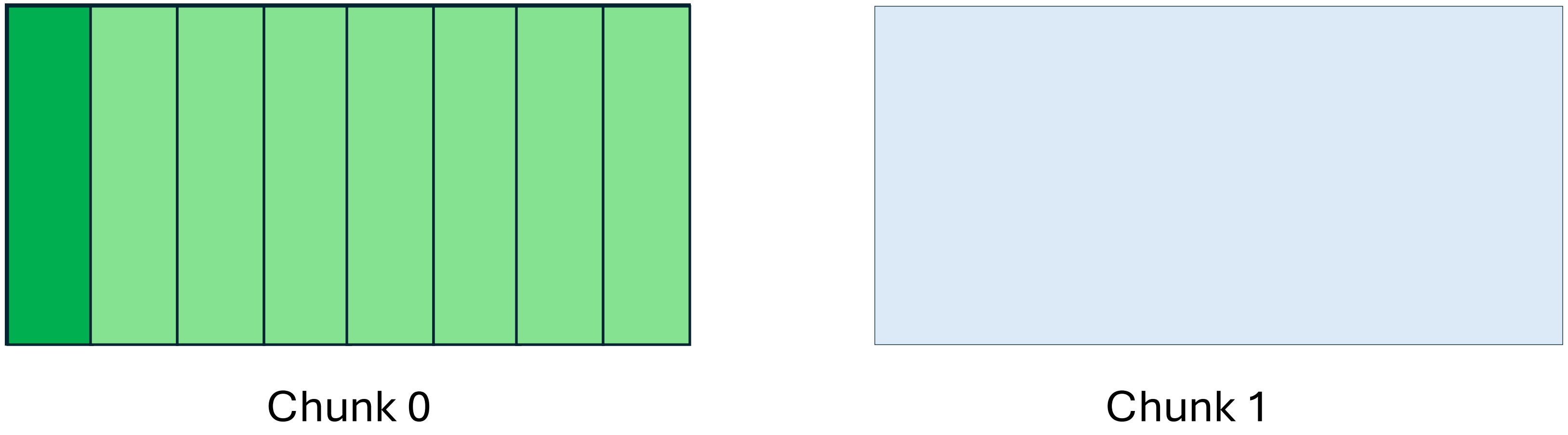
Method

Compute passage embeddings

Chunks are loaded into RAM

Method

Computing passage embeddings

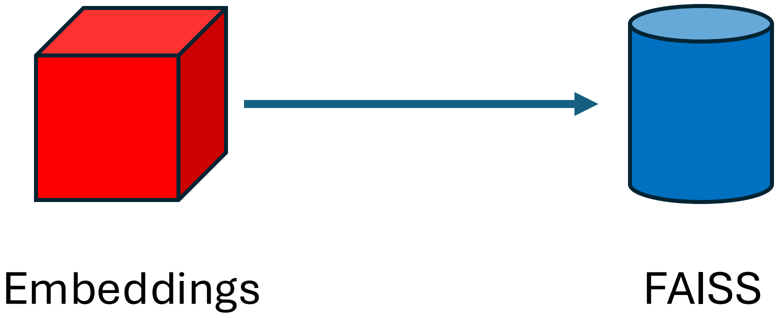


Batches are processed by GPU
(SBERT encoding)

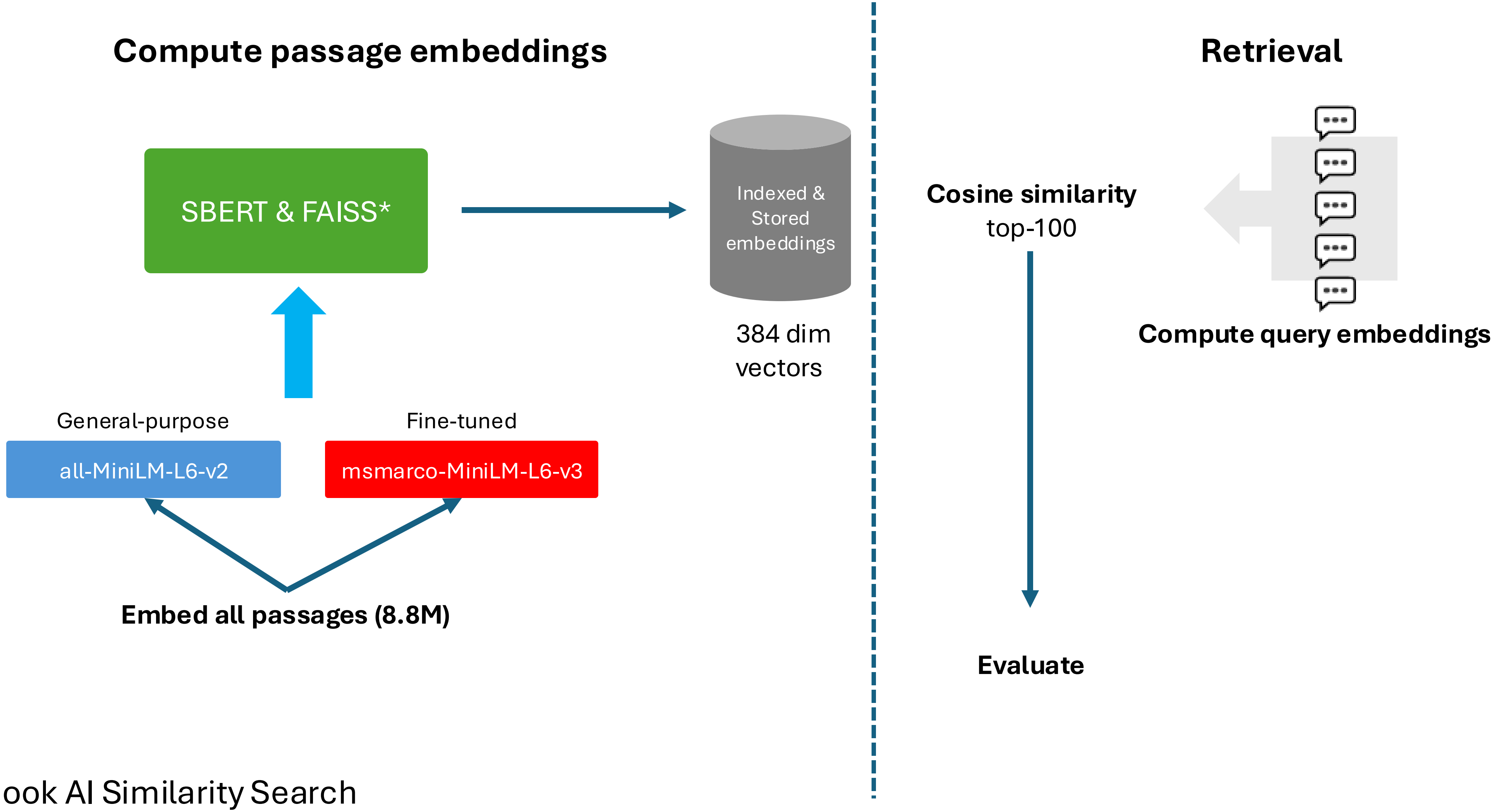
Method

Facebook AI Similarity Search (FAISS)

- Efficient **similarity search**
- Can **find k most similar** vectors in a very large set (millions or billions)
- Offers different indexing types, we use **IndexFlatIP (exact search using inner product)**
- Works on **CPU and GPU** (Linux only)



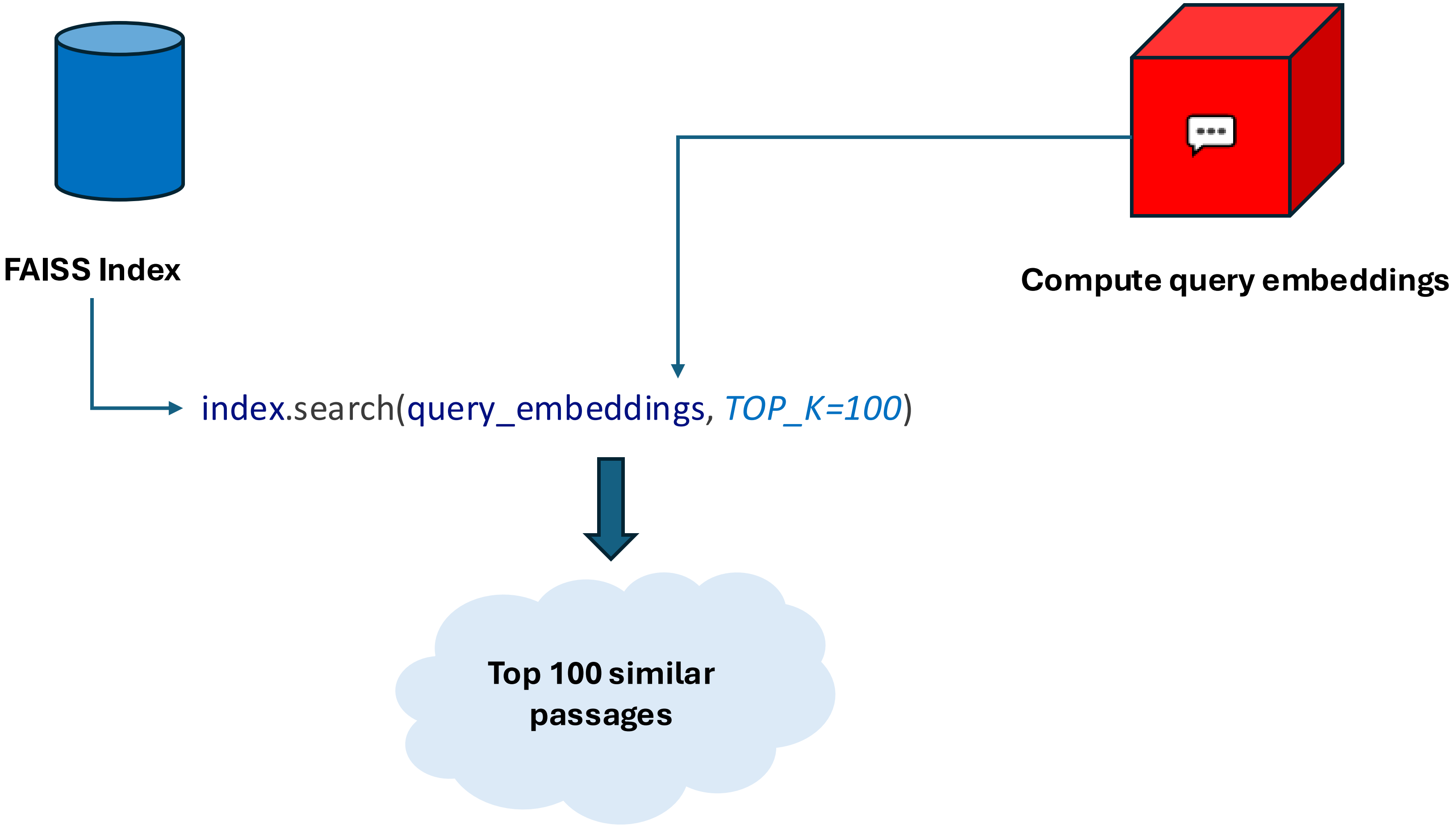
Method



* Facebook AI Similarity Search

Method

Retrieval



Method

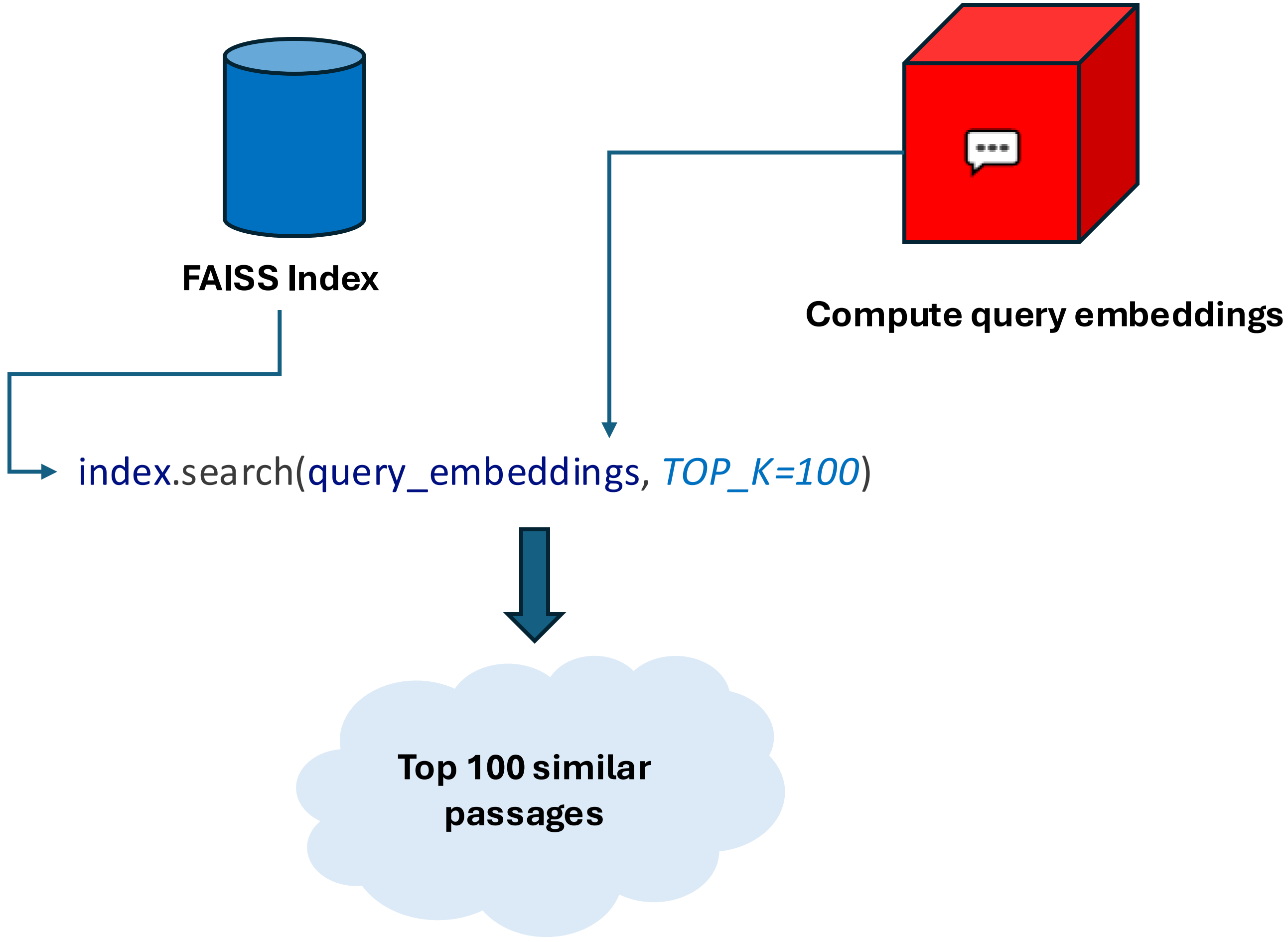
Evaluation metrics

- Mean reciprocal rank (MRR) @ rank

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\mathrm{rank}_i}.$$

Q = queries

Ideally the rank would be 1 for each query



Agenda

- Goal
- Dataset
- Related Work
- Method
- Results

Results

Previous mentioned research

Single stage:

- **BM25** *MRR@10 = 16.7*

Two stage methods:

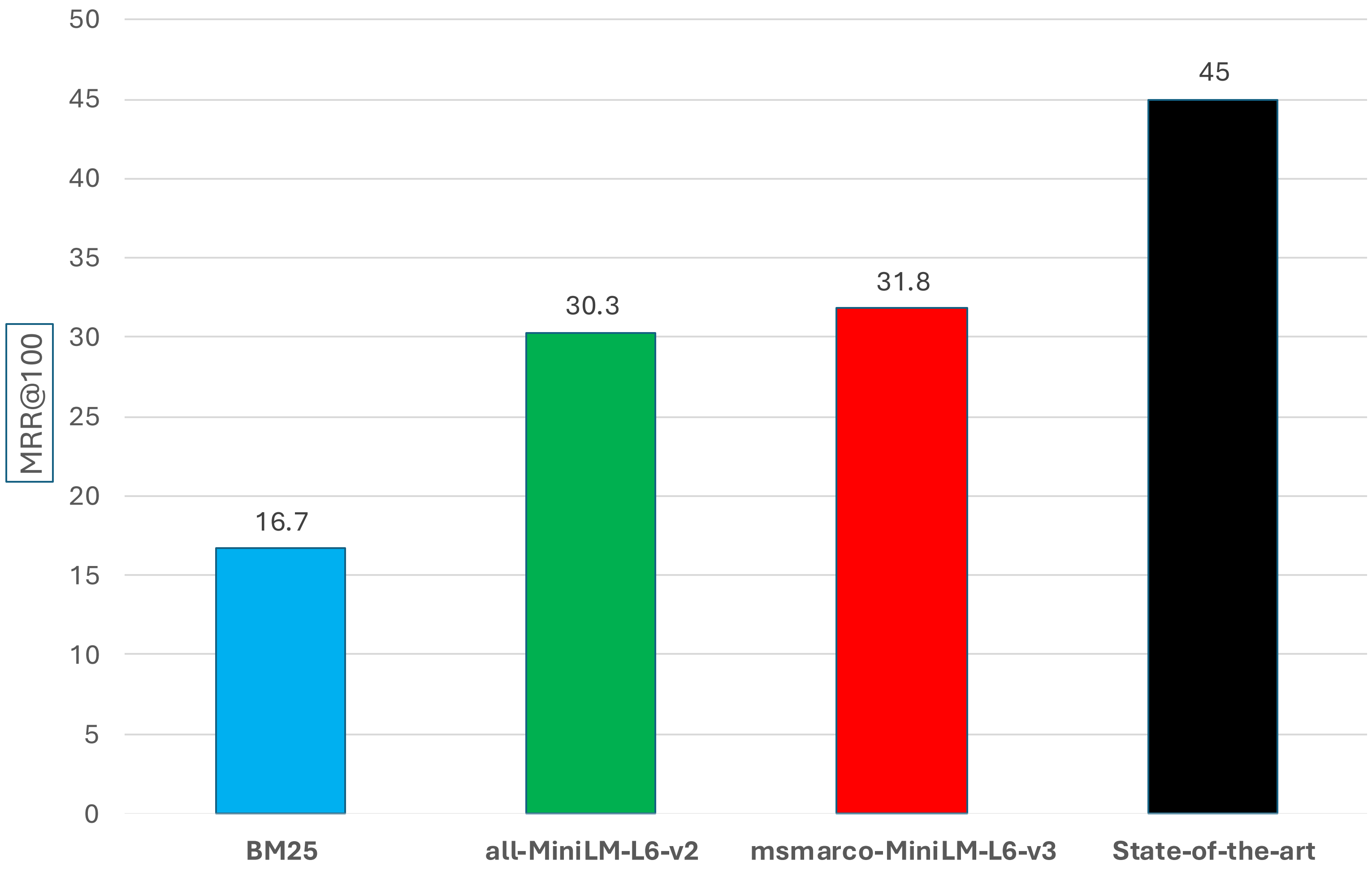
- **BM25 + BERT base** *MRR@10 = 34.7*
- **BM25 + BERT large** *MRR@10 = 36.5*

<u>Our implementation</u>

- **Fine-tuned model** *MRR@10 = 31.8*
- **General purpose model** *MRR@10 = 30.3*

Other three stage methods:

- *MRR@10 = 39-45*



Conclusions and future work

Conclusion

- **Breakthrough Achievement**
  - sBERT with FAISS secures an MRR@10 of 32, decisively surpassing BM25 (16.7) on MS MARCO.
  - Dense retrieval redefines precision, eclipsing traditional lexical approaches.
- **Superior Optimization**
  - Fine-tuned msmarco-MiniLM-L6-v3 (MRR@10: 32) outperforms allMiniLM-L6-v2 (MRR@10: 30)
- **Paradigm Shift**
  - Neural embeddings paired with scalable indexing establish a new benchmark in information retrieval excellence

Future work

- **Expanded Scope**
  - Larger datasets will amplify model robustness and impact.
- **Refined Calibration**
  - Advanced fine-tuning could elevate accuracy to higher levels. Larger embedding vector dimension.
- **Optimized Efficiency**
  - Enhanced indexing techniques will streamline retrieval speed

Questions?