

On the Theoretical Limitations of Embedding-Based Retrieval

– Paper Presentation and Reproduction –

Gábor Hosu

Faculty of Mathematics and Computer Science
Babeş-Bolyai University

2026-02-02

Contents

1. Problem Statement and Related Work

2. Representational Capacity of Vector Embeddings

3. Empirical Connection

3.1 Best-Case Optimization

3.2 Real-World Datasets

4. Conclusion

Problem Statement and Related Work

Problem Statement and Related Work

Problem

- Vector embeddings still follow the single-vector paradigm
- Retrieval failures: poor data, unrealistic queries, or embedding-space geometry?

Related Work

- IR evolution: parse vectors \rightarrow dense embeddings
- Task complexity: QUEST (logical ops), BRIGHT (Leetcode reasoning)
- Empirical findings: lower dimensions \Rightarrow more false positives; affects bias-variance tradeoff

Capacity of Vector Embeddings

Representational Capacity of Vector Embeddings

- Embedding representation:

$$\text{query}_i \xrightarrow{\text{emb. model}} u_i \in \mathbb{R}^d, \quad \|u_i\| = 1, i \in \{1, \dots, m\}$$

$$\text{doc}_j \xrightarrow{\text{emb. model}} v_j \in \mathbb{R}^d, \quad \|v_j\| = 1, j \in \{1, \dots, n\}$$

- $\text{cosine-sim}(u_i, v_j) = u_j^T v_j$ – dot product as matrix multiplication
- Query-relevance (qrel) matrix: $A = [\text{doc}_j \text{ relevant to query}_i]_{ij} \in \{0, 1\}^{m \times n}$
- **Q:** When is retrieval accurate?
- **A:** When all top- k results are returned.

Representational Capacity of Vector Embeddings

$$U := \left[\begin{array}{ccc} & | & \\ \cdots & u_i & \cdots \\ & | & \end{array} \right]_i \in \mathbb{R}^{d \times m} \quad V := \left[\begin{array}{ccc} & | & \\ \cdots & v_j & \cdots \\ & | & \end{array} \right]_j \in \mathbb{R}^{d \times n}$$

- \Rightarrow the similarity matrix

$$\mathbb{R}^{m \times n} \ni B := [u_i^T v_j]_{ij} = \left[\begin{array}{ccc} \vdots & & \\ - & u_i^T & - \\ \vdots & & \end{array} \right] \left[\begin{array}{ccc} & | & \\ \cdots & v_j & \cdots \\ & | & \end{array} \right] = U^T V$$

Representational Capacity of Vector Embeddings

- **Q:** When is retrieval accurate?
- **A:** When all top- k results are returned.



$$\forall i, j, k \quad A_{ij} > A_{ik} \implies B_{ij} > B_{ik}, \quad (\text{rop})$$

i.e. the row-wise order is preserved.

Representational Capacity of Vector Embeddings

- **Q:** What embedding dimension preserves row-wise ordering?
- **A:** The minimal (embedding) dimension d s.t. for $B \in \mathbb{R}^{m \times n}$

$$\exists U \in \mathbb{R}^{d \times m} \text{ and } V \in \mathbb{R}^{d \times n} \text{ s.t. } B = U^T V \quad (\text{fact})$$
$$\text{and (rop).}$$

- **Q:** Minimal d via linear algebra?
- **A:** $d_{\min} = \text{rank} B$
- So we can summarize (fact) and (rop) as

$$\text{rank}_{\text{rop}} A := \min \{ \text{rank} B \mid B \in \mathbb{R}^{m \times n}, \text{ s.t. } \forall i, j, k \quad A_{ij} > A_{ik} \implies B_{ij} > B_{ik} \},$$

the **row-wise order-preserving rank** of the qrel matrix A .

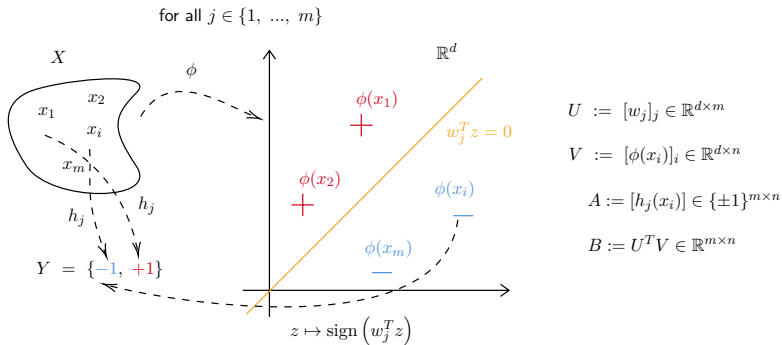
Representational Capacity of Vector Embeddings

Q: How to calculate $\text{rank}_{\text{rop}} A$?

Idea: Quantify how hard it is to separate relevant from irrelevant documents.

Representational Capacity of Vector Embeddings

Learning theory



- How complex is a given binary classification problem?
- Complexity = min. dimension d s.t. $\text{sign} B = A$ and (fact):

$$\text{rank}_{\pm} A := \min \{ \text{rank} B \mid B \in \mathbb{R}^{m \times n} \text{ s.t. } \text{sign} B = A \} - \text{sign-rank of } A$$

Representational Capacity of Vector Embeddings

By Warren's theorem in real algebraic topology, ...

Lemma

Let $r < N/2$.

Then $\#(N \times N \text{ sign-matrices of sign-rank } \leq r)$ does not exceed $2^{O(rN \log N)}$.

Observation

$$\#(N \times N \text{ sign-matrices}) = 2^{N^2} > 2^{O(rN \log N)} \implies$$

$\exists N \times N \text{ sign-matrices with sign-rank } > r \text{ for large } N.$

So there are sign-matrices with arbitrary large sign-rank.

Representational Capacity of Vector Embeddings

The key result from the paper:

Theorem

Let $A \in \{0, 1\}^{m \times n}$ be a binary matrix. Then $2A - \mathbf{1}_{m \times n} \in \{\pm 1\}^{m \times n}$, and we have

$$\text{rank}_{\pm}(2A - \mathbf{1}_{m \times n}) - 1 \leq \text{rank}_{\text{rop}} A \leq \text{rank}_{\pm}(2A - \mathbf{1}_{m \times n}).$$

Observations

- Sign-rank computation is NP-hard.
- Lemma: $\min d$ can be arbitrarily large compared to $\text{rank} A$.
- Fixed $d \implies$ some qrel matrices are not representable.
- Order-preserving embedding in $d \implies$ bounded sign-rank (estimable via optimization).

Empirical Connection

Best-Case Optimization

- Theory: embedding capacity is geometric, not linguistic.

Free embedding optimization

- All $\binom{n}{2}$ queries over n documents – dense qrels
- For fixed dimension d , find largest representable n – critical- n
- Optimize query and document embeddings via gradient descent
- Perfect fit \equiv 100% recall
- Contrastive loss: pull relevant docs closer than irrelevant ones

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{|Q|} \sum_{q \in Q} \sum_{\substack{d' \in D \\ d' \text{ relevant to } q}} \log \frac{\exp(\text{sim}(q, d')/\tau)}{\sum_{d \in D} \exp(\text{sim}(q, d)/\tau)}$$

Best Case Optimization

Implementation

- Original paper: training with JAX on H100 GPUs and TPU v5's
- Reproduced: training with PyTorch on a Kaggle P100 GPU
- Optimizer: Adam
- Training hyperparams:

```
1 def train(  
2     num_of_docs: int,  
3     dimension: int,  
4     max_patience: int = 1000,  
5     temp: float = 0.1,  
6     learning_rate: float = 0.01,  
7     max_iters: int = 100000,  
8     min_delta: float = 0.00001  
9 ) -> float:  
10     ...
```


Best Case Optimization

PyTorch Model:

```
1 class FreeEmbeddingsModel(torch.nn.Module):
2     # ... init attributes ...
3     def forward(self):
4         """During training the embeddings must be normalized after optimization"""
5         self.__qrel_matrix = self.__qrel_matrix.to(self.docs.device)
6
7         queries_norm = self.queries
8         docs_norm = self.docs
9
10        logits = (queries_norm @ docs_norm.T) / self.__temp
11        log_probs = torch.log_softmax(logits, dim=1)
12
13        sum_pos_log_probs = (log_probs * self.__qrel_matrix).sum()
14        M = self.__qrel_matrix.sum()
15        total_loss = -sum_pos_log_probs / M
16        return total_loss
```

Best Case Optimization

Results

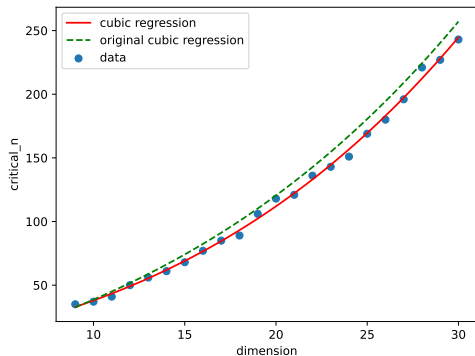


Figure: Free embedding optimization results (reproduced up to dim. 30; paper reports dim. 45)

$$y_o = 0.0037d^3 + 0.0520d^2 + 4.0309d - 10.5322$$

$$y_r = 0.003866d^3 + 0.06175d^2 + 2.84d - 0.4842$$

$$R_o^2 = 0.999 \quad R_r^2 = 0.997$$

$$\text{RSSE} = 58.287 \quad \text{RMSE} = 8.243$$

Emb. dimension d	Critical n
512	5.37×10^5
768	1.79×10^6
1024	4.22×10^6
4096	2.67×10^8

Table: Extrapolated values based on the reproduced regression.

Real-World Datasets

$$\rho = \frac{|E|}{\frac{|V|(|V|-1)}{2}}$$

$$s_i = \sum_{j \in N(i)} w_{ij}$$

$$\bar{s} = \frac{1}{|V|} \sum_{i \in V} s_i$$

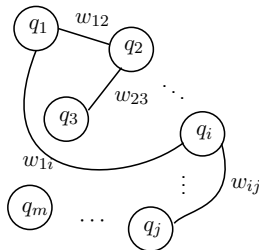


Figure: Query-query Jaccard-weighted graph with graph density and average strength metrics.

- Test on real-world datasets
- Existing benchmarks are too sparse

Dataset Name	Graph Density	Average Query Strength
NQ	0	0
HotPotQA	0.000037	0.1104
SciFact	0.001449	0.4222
FollowIR Core17	0.025641	0.5912

Table: Dataset statistics on standard benchmark models.

Real-World Datasets – LIMIT

LIMIT dataset's original construction:

- **Document structure:** X likes $attr_1, \dots, attr_m$
- **Query structure:** Who likes $attr_i$?
- **Relevance pattern:** $\binom{n}{2} \times n$
 - 46 documents, $\sim \binom{46}{2} \approx 1000$ queries
- Names from public datasets
- Attributes generated via Gemini 2.5 Pro + BM25 filtering
- Small and large-scale versions (up to 50k docs)
- **Dataset statistics:** 0.085481 density, 28.4653 average query strength

Real-World Datasets - Benchmark on LIMIT

Models from the paper:

Model	Parameters	MRL
Snowflake Arctic L	0.3B	✓
E5-Mistral 7B	7B	✗
GritLM 7B	7B	✗
Qwen3 Embed	8B	✓
Promptriever Llama3 8B	8B	✗
Gemini Embed	unknown	✓

Table: Traditional SoTA embedding models.

Other models:

- BM25
- GTE-ModernColBERT

Inference device: A100 GPU

Real-World Datasets - Benchmark on LIMIT

Models from the reproduction:

Model	Parameters	MRL
Snowflake Arctic L	0.3B	✓
E5-Mistral 7B	7B	✗
GritLM 7B	7B	✗
Qwen3 Embed	8B	✓
Promptriever Llama3-8B	8B	✗
Gemini Embed	unknown	✓
Qwen3 Embed	0.6B	✓

Table: Traditional SoTA embedding models

Other models:

- BM25
- GTE-ModernColBERT

Inference devices:

- Kaggle T4 GPUs with 4-bit quantization using *bnb-my-repo*.
- GeForce GTX 1650

Real-World Datasets - Benchmark on LIMIT

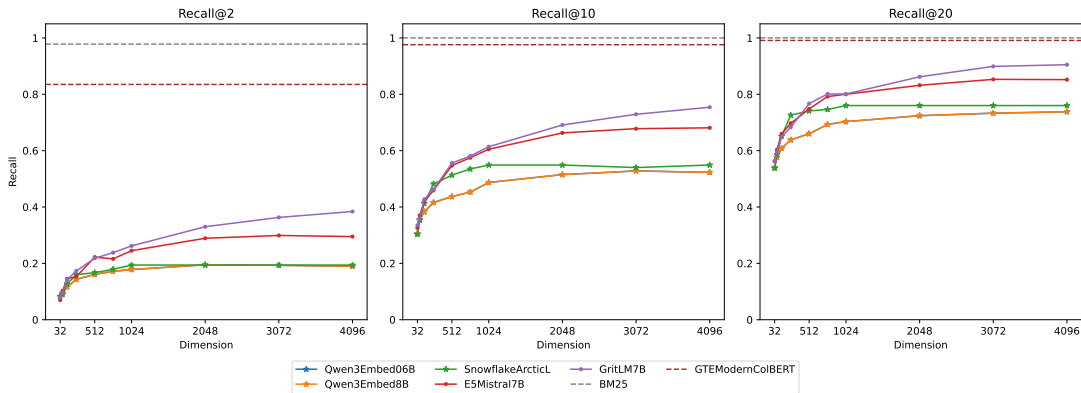


Figure: Expected results (for selected models) on LIMIT-small.

Real-World Datasets - Benchmark on LIMIT

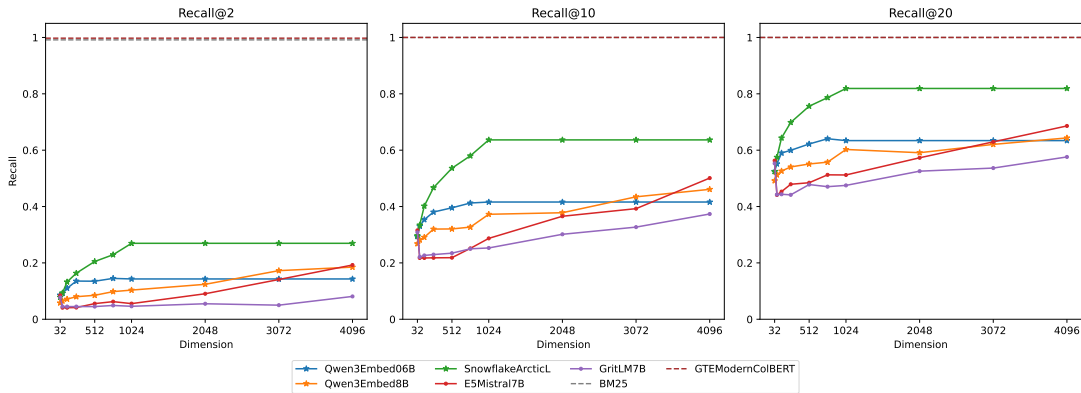
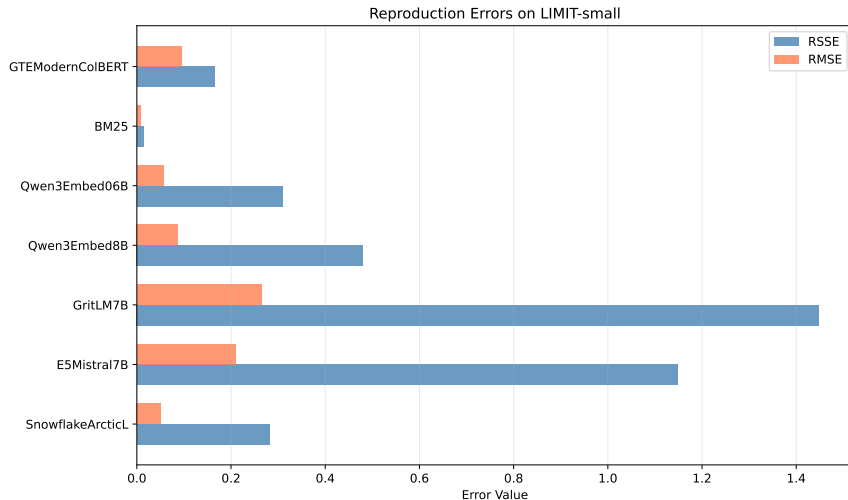


Figure: Reproduced results on LIMIT-small.

Real-World Datasets - Benchmark on LIMIT



Real-World Datasets - Domain shift?

- **Q:** Do SoTA models fail because the task is hard, or because the dataset is too specific?
- **A:** Fine-tune a SoTA model on a similar task and evaluate it on LIMIT
- **Issue:** LIMIT has only a test set \implies we must build a training set with the same name distribution

Real-World Datasets - Domain shift?

Training split construction:

- Paper: size-matched, manually created
- Ours: replicated test pipeline
 - GPT-5 (public) for attribute generation with prompting
 - *all-mpnet-base-v2* embeddings instead of BM25 for semantic filtering
 - Original data generation otherwise unchanged

	Test attributes	Train attributes
#attributes	1850	1857
Inner similarity	0.668	0.652
Cross similarity	0.264	

Table: Attribute counts and cosine similarities within and sets.

Real-World Datasets - Domain shift?

Fine-tuning:

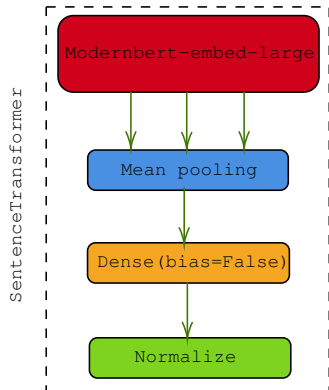


Figure: Architecture of the fine-tuned model.

- Derived model fine-tuned on both sets
- Training parameters:
 - Learning rate 5e-5
 - 5 epochs, max batch size (2000)
 - BatchSamplers.NO_DUPLICATES
 - Paper: MultipleNegativesRankingLoss
 - Ours: CachedMultipleNegativesRankingLoss with mini_batch_size=8
- Devices:
 - Paper: A100 GPU
 - Ours: 1×T4 Kaggle GPU

Real-World Datasets - Domain shift?

Fine-tuning results:

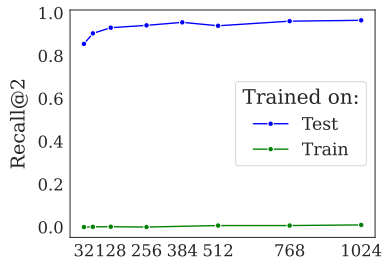


Figure: Expected results from the paper.

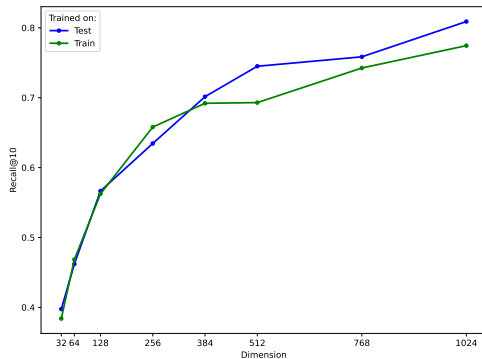


Figure: Reproduced results.

Real-World Datasets - Qrel patterns

- Dense qrel patterns in LIMIT make retrieval more difficult.
- Intuitively, sparser patterns should be easier for models to learn.
- **Q:** Can we show this empirically?

Real-World Datasets - Qrel patterns

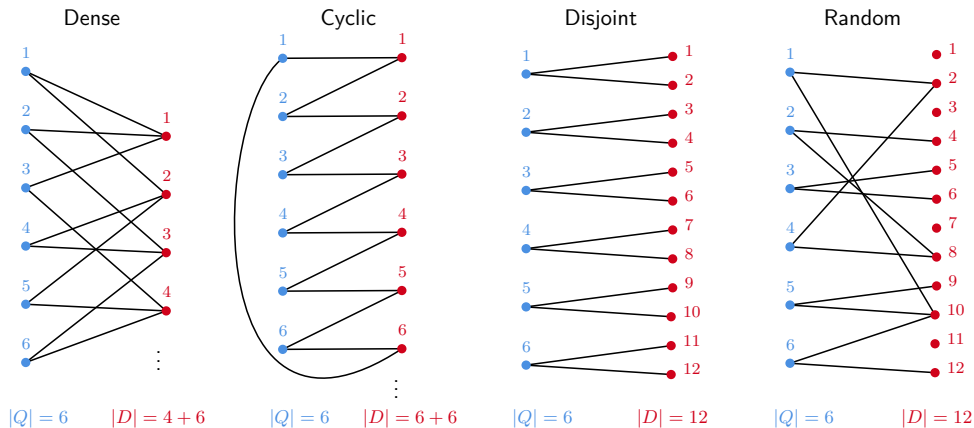


Figure: Different qrel patterns with 6 queries, 12 docs, $k = 2$ relevant doc/query.

Real-World Datasets - Qrel patterns

Experimental setup:

- Evaluated on previously introduced qrel patterns and LIMIT models
- Two settings:
 - **Paper:** $1000 \approx \binom{46}{2}$ queries, 50000 docs ($46 + 49954$); A100 GPU
 - **Reproduction:** $23 \approx \binom{8}{2}$ queries, 2000 docs ($8 + 1992$); P100 & GeForce GTX 1650 GPU
- $k = 2$ relevant documents per query (both settings)

Real-World Datasets - Qrel patterns

Results:

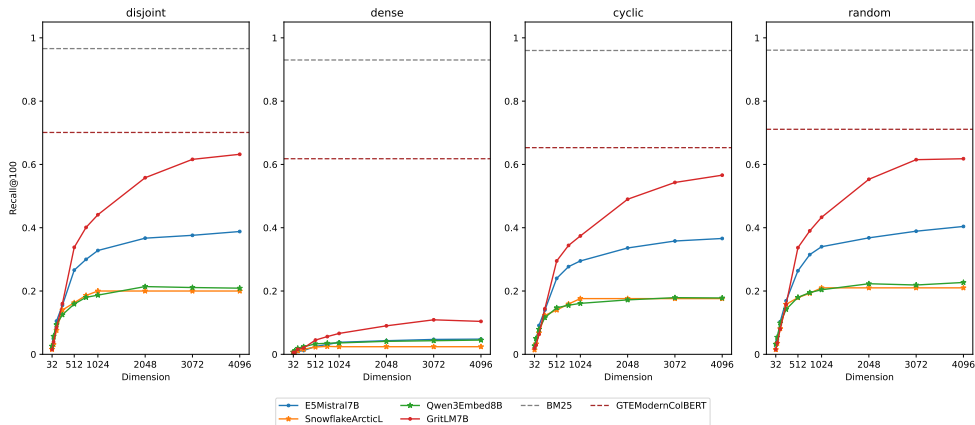


Figure: Expected results from the paper.

Real-World Datasets - Qrel patterns

Results

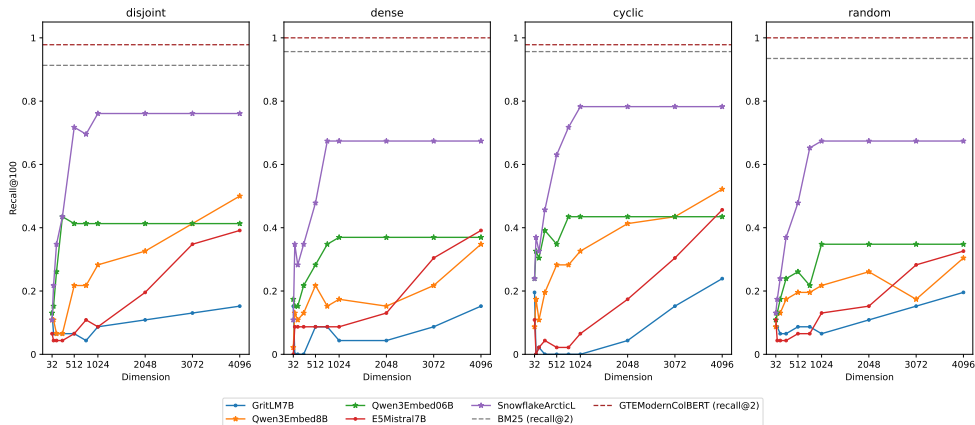


Figure: Reproduced results (quantized larger models).

Real-World Datasets - BEIR vs LIMIT

- BEIR – Benchmarking Information Retrieval – benchmark datasets
- Models perform well \iff overfit on BEIR
- **Q:** Are BEIR and LIMIT connected?

Real-World Datasets - BEIR vs LIMIT

- **A: No**, the sample correlations are not statistically significant.

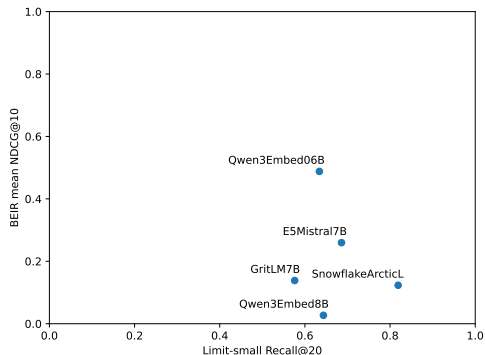


Figure: SciFact and NFCorpus datasets from BEIR vs Limit-small (on quantized models):
 $r = -0.162$, $p\text{-value} = 0.793 \gg 0.05$.

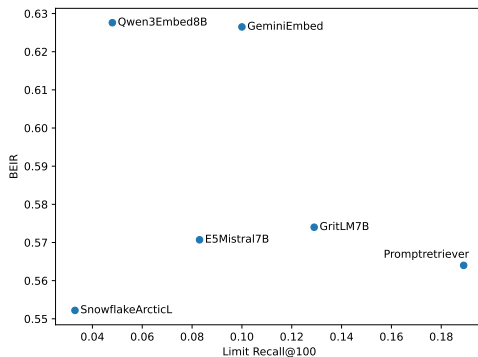








Figure: BEIR vs Limit from the paper:
 $r = -0.208$, $p\text{-value} = 0.691 \gg 0.05$.

Conclusion

Key Takeaways

- Single-embedding retrieval is limited by embedding space geometry.
- **Ways to overcome this:**
 - Higher-dimensional embeddings or traditional statistical methods (BM25, TF-IDF).
 - Multi-embedding models (e.g., ModernColBERT) to preserve token-level information.

References

-  Weller, O., Boratko, M., Naim, I., & Lee, J. *On the Theoretical Limitations of Embedding-Based Retrieval*. Google DeepMind, 2025. arXiv:2508.21038
-  Hosu, G. *embedding_dimension_limit*, 2025. GitHub repository. [Link to repository](#)
-  Alon, N., Moran, S., & Yehudayoff, A. *Sign rank versus VC dimension*. arXiv preprint arXiv:1503.07648, 2015. arXiv:1503.07648
-  Thakur, N., Reimers, N., Rüchlé, A., Srivastava, A., & Gurevych, I. *BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models*, 2021. arXiv:2104.08663
-  van den Oord, A., Li, Y., & Vinyals, O. *Representation Learning with Contrastive Predictive Coding*, 2019. arXiv:1807.03748
-  Gutmann, M., & Hyvärinen, A. *Noise-contrastive estimation: A new estimation principle for unnormalized statistical models*, 2010. [Link to PDF](#)