HEARTS: Hypergraph-based Related Table Search

Allaa Boutaleb, Alaa Almutawa, Bernd Amann, Rafael Angarita, and Hubert Naacke
Sorbonne Université, CNRS, LIP6, F-75005 Paris, France {**firstname.lastname@lip6.fr**}

1. Introduction

Recent related table search methods leverage tabular representation learning and language models to encode tables into vector representations for efficient semantic search. The main challenge of these models is to retain essential structural properties of tabular data. Graph-neural networks have shown to be efficient in solving certain challenges like row/column permutation sensitivity and multi-table representation. In this context, we present HEARTS, a related-table search system powered by HyTrel [1], a hypergraph-enhanced Tabular Language Model (TaLM). By representing tables as hypergraphs with cells as nodes and rows, columns, and tables as hyperedges, HyTrel preserves relational properties such as row and column order invariance, making it a robust solution for related table search tasks.

2. Key Questions

Table representation: How do Graph-Based Table Language Models compare to BERT-based methods for related table search?

Search Efficiency versus Precision: How does simple index-based similarity search compare to task tuned table matching algorithms for related table search?

Benchmarks: Do existing benchmarks sufficiently challenge related table search methods and reflect performance on real-world data?

Hybrid Search: What is the effect of integrating set-based equi-join similarity with vector-based semantic similarity for joinable column discovery?

3. HEARTS

Preprocessing:

- Table columns are vectorized using a pre-trained hypergraph-enhanced TaLM (HyTrel).

Joinable table search:

- *Offline:* flat FAISS column index creation.
- *Online: Column* index search + LSH Ensemble filtering & re-ranking.

Unionable table search :

- **Column Clustering:**
  - ○ *Offline:* UMAP dimensionality reduction + HDBSCAN clustering.
  - ○ *Online:* Jaccard similarity of column cluster assignments.
- **Per-table Column Pooling :**
  - ○ *Offline:* Per table mean/max pooling of column vectors → flat FAISS table index creation.
  - ○ *Online:* Table index search.

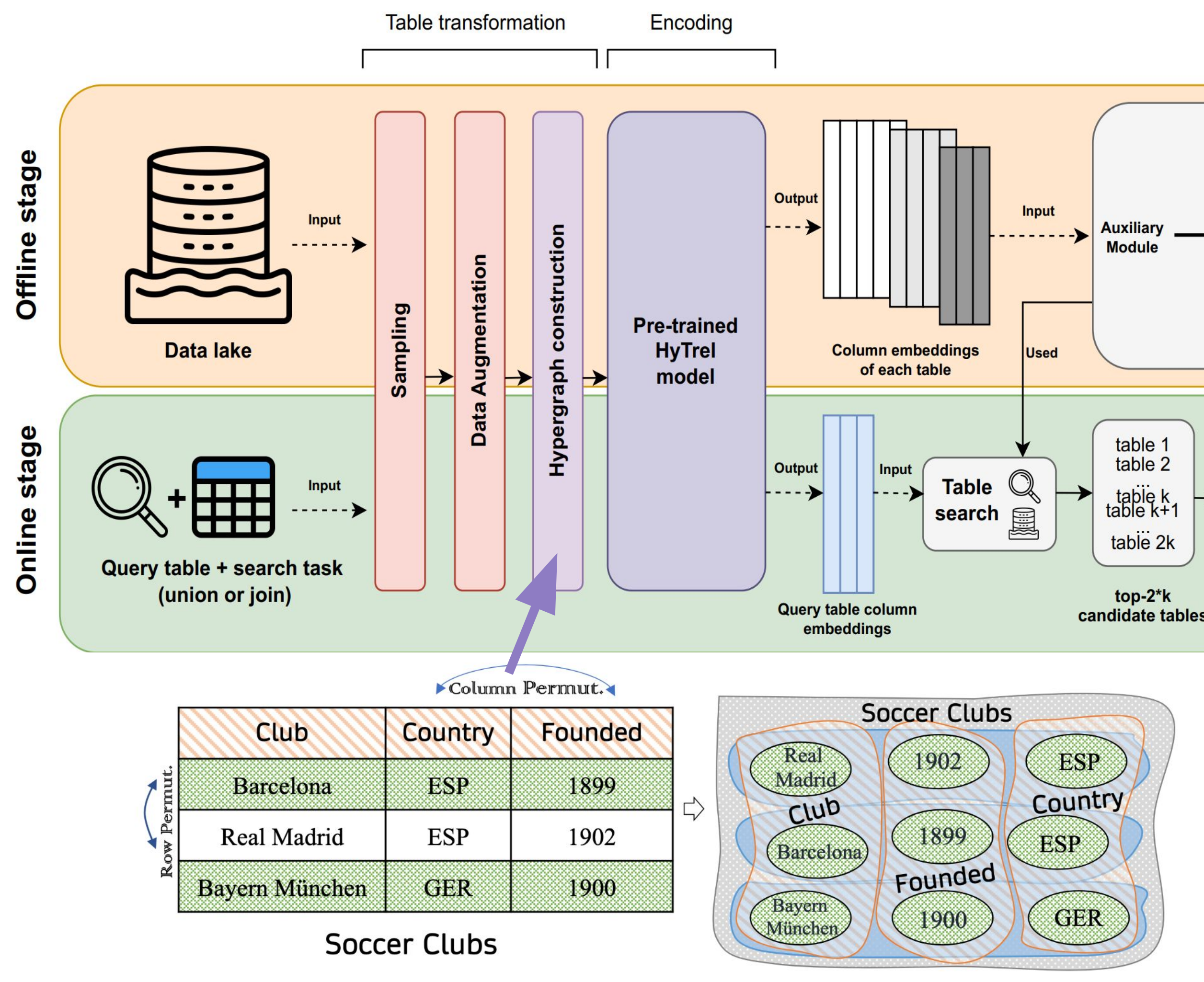


Figure taken from [1]

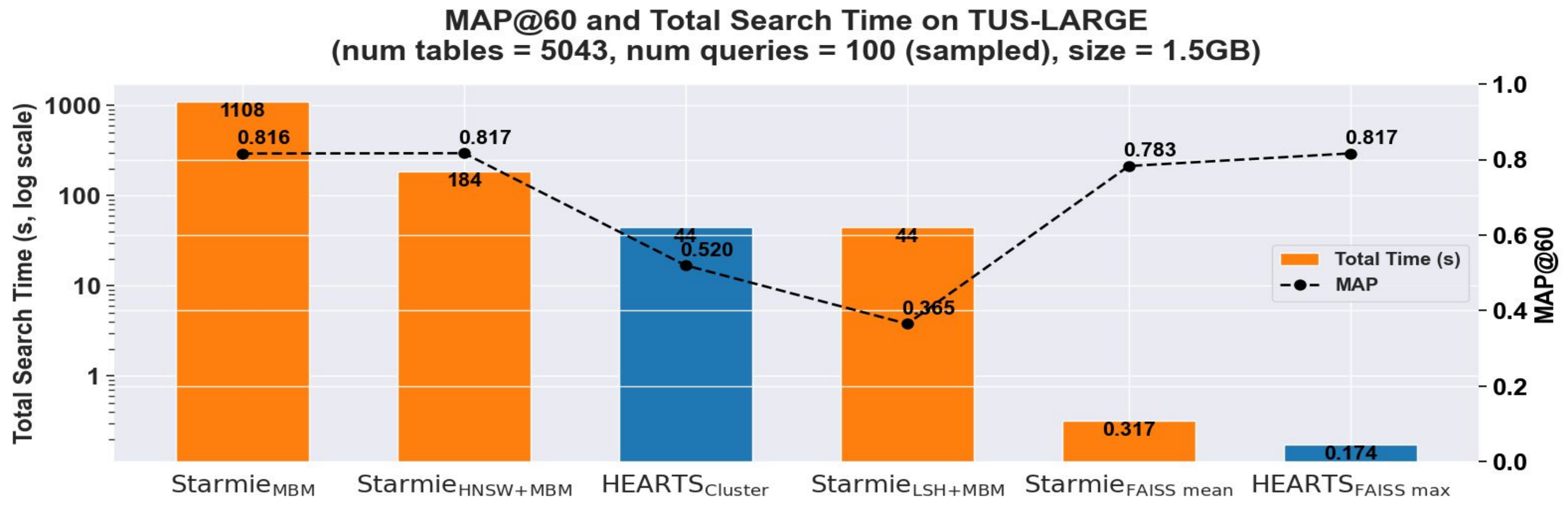
Figure 1: An example of modeling a table as a hypergraph. Cells make up the nodes and the cells in each row, column, and the entire table form hyperedges. The table caption and the header names are used for the names of the table and column hyperedges. The hypergraph keeps the four structural properties of tables, e.g., the invariance property of the table as the row/column permutations result in the same hypergraph.

4. Unionable Table Search Results

- **Benchmarks:** SANTOS [6], TUS and TUS-LARGE [5].
- **Starmie [2]:** RoBERTa (10 epochs), contrastive learning, MBM (Hungarian algorithm), LSH/HNSW.
- **HEARTS (Ours):** HyTrel embeddings (pretrained with contrastive loss), 30 rows & 20 columns sampled per table.
- For FAISS search, **Starmie** and **HEARTS** use **mean and max pooling** resp.

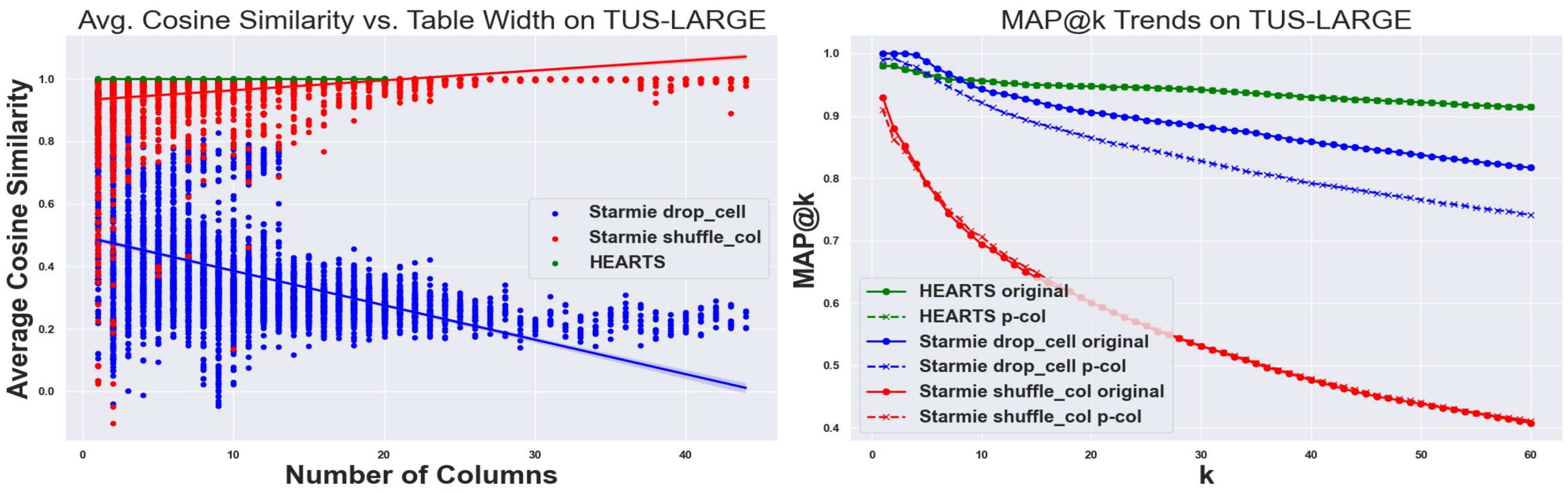
Let $T(k) = \{t_1, \ldots, t_k\}$ be the top k retrieved items, R the set of relevant items and Q the queries (tables or columns).

$$\mathbf{P@k} = \frac{|T(k) \cap R|}{k}, \quad \mathbf{R@k} = \frac{|T(k) \cap R|}{|R|}, \quad \mathbf{AP@k} = \frac{1}{\min(k, |R|)} \sum_{j=1}^{k} \mathbf{1}_{\{t_j \in R\}}\, \mathbf{P@j}, \quad \mathbf{MAP@k} = \frac{1}{Q} \sum_{q=1}^{Q} \mathbf{AP^{(q)}@k}$$



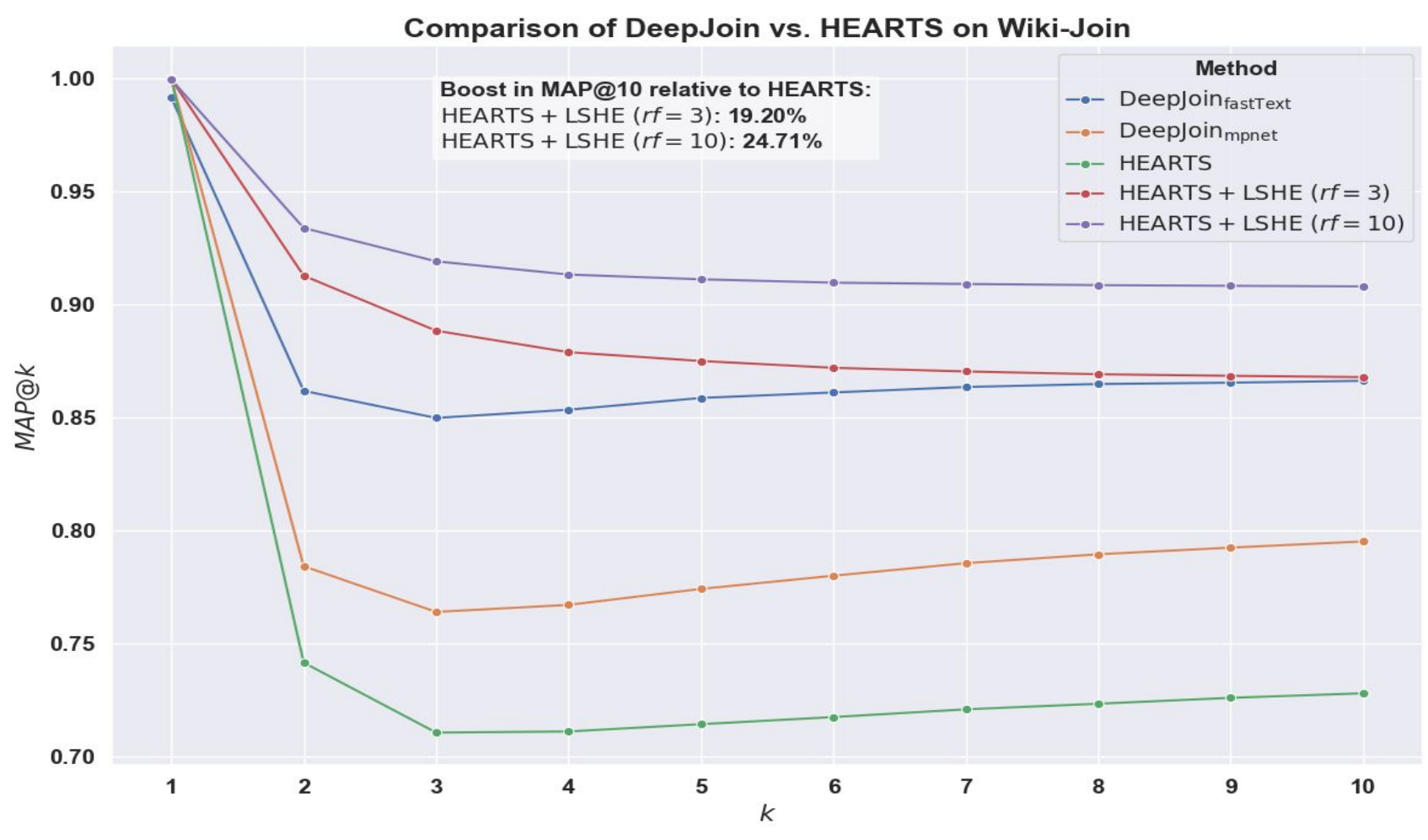
	SANTOS (MAP@10)		TUS (MAP@60)		TUSLARGE (MAP@60)	
Search Method	HEARTS	STARMIE	HEARTS	STARMIE	HEARTS	STARMIE
MBM	0.909 (33.1s)	0.948 (54.5s)	0.960 (253.5s)	0.830 (413.6s)	0.914 (508.2s)	0.816 (1108.8s)
Cluster	0.962 (23.9s)	0.983 (31.7s)	0.569 (25.4s)	0.664 (36.0s)	0.520 (45.0s)	0.679 (87.2s)
FAISS	0.858 (0.02s)	0.982 (0.04s)	0.898 (0.06s)	0.755 (0.13s)	0.817 (0.17s)	0.783 (0.32s)
HNSW	/	0.948 (38.5s)	/	0.830 (97.1s)	/	0.817 (184.9s)
LSH	/	0.936 (12.4s)	/	0.342 (28.7s)	/	0.365 (44.8s)

Robustness against column-shuffling: **Starmie** with cell dropping ***(drop_cell)*** or column-shuffling ***(shuffle_col)*** augmentation versus **HEARTS**.



5. Joinable Table Search Results

- **Benchmark**: Wiki-Join [4] *(MAP@10)*.
- **DeepJoin [3]** variants without fine tuning: fastText and mpnet (all-mpnet-base-v2).
- **HEARTS** with **LSH Ensemble** (128 permutations and 32 partitions) :
  - ○ **RF = 3**: retrieve 3x10 initial candidates.
  - ○ **RF = 10**: retrieve 10x10 initial candidates.



6. Conclusion & Perspectives

- Pretrained Table Language Models (HyTrel) achieve performance comparable to existing BERT-based methods *without requiring any additional fine-tuning*.
- *Fast similarity search* methods yield near-identical performance compared to sophisticated matching algorithms for unionable table search which highlights the need for more challenging benchmarks and evaluation scenarios.
- Combining set-based equi-join similarity (LSH Ensemble) and vector-based fuzzy-join similarity improves joinable column retrieval.

References

[1] P. Chen et al. *Hytrel: Hypergraph-enhanced tabular data representation learning*. NeurIPS, 2024.
[2] G. Fan et al. *STARMIE: Semantics-aware dataset discovery from data lakes*. PVLDB, 2023.
[3] Y. Dong et al. *DeepJoin: Joinable table discovery with pre-trained language models*. IEEE TKDE, 2023.
[4] K. Srinivas et al. *LakeBench: Benchmarks for data discovery over data lakes*. arXiv, 2023.
[5] F. Nargesian et al. *TUS: Table union search on open data*. PVLDB, 2018.
[6] A. Khatiwada et al. *SANTOS: Relationship-based semantic table union search*. SIGMOD, 2023.

Code available on GitHub

github.com/Allaa-boutaleb/HEARTS