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

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 [10] and multi-table representation [6]. In this context, we present HEARTS[1], 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.

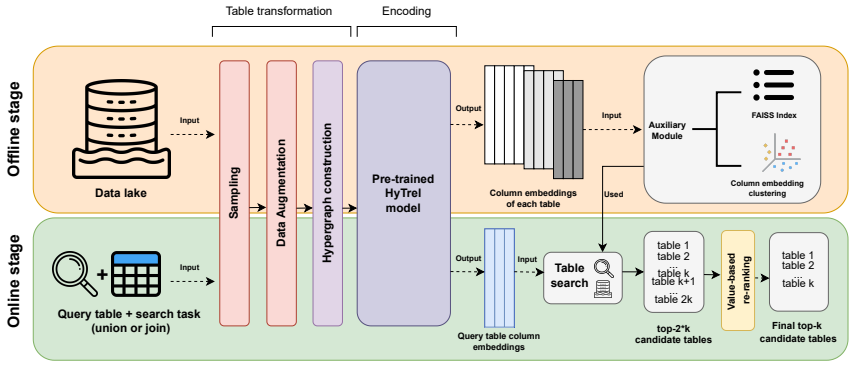

Fig. 1. HEARTS pipeline overview

Figure 1 illustrates the overall architecture of our system. The **offline stage** prepares data lake tables by sampling representative rows and addressing missing metadata, such as null column names. It then transforms the processed tables into hypergraphs and generates column embeddings using a HyTrel checkpoint pre-trained with a contrastive objective. The obtained table column vector embeddings are stored in a FAISS index [7,4], either as individual vectors for join search or as aggregated table-level vectors for union search. Alternatively to FAISS indexing, in our experiments we also explore a column clustering approach where tables are described by a set of semantic column clusters. The **online**

[1] Code available at `https://github.com/Allaa-boutaleb/HEARTS`

stage processes a query table to generate column embeddings using the same table transformation and encoding steps as in the offline stage. The FAISS index then returns the top-k column neighbors of the query column (join) or aggregated table (union) embedding. Alternatively, unionable tables are identified by the Jaccard similarity of their semantic column clusters. The **post-processing stage (for join search)** improves the quality of results by combining semantic and *value-based* similarity [2]. LSH Ensemble and MinHash sketches [14] refine the candidate table list by measuring actual value overlap, reducing false positives from semantic matching alone.

Evaluation We evaluate our approach on multiple public benchmarks, including SANTOS [8] (550 tables) and TUS [11] (up to 5K tables) for union search, as well as Wiki-Join [9,12] for join search. We explore different unionable table search strategies, including clustering-based search, achieving comparable accuracy to exact bipartite graph matching [5] with better efficiency. We also show that Faiss indexes, using column vectors directly for join search and max-pooled compressed table vectors for union search, achieve 99.99% query speedup with minimal precision/recall degradation compared to exact methods. Other key findings include:

- Competitive performance compared to state-of-the-art methods like Starmie [5] for union search and DeepJoin [3] for join search, with stable memory requirements even as data lake size varies.
- Strong preservation of structural properties (row and column order invariance), demonstrating advantages over BERT-based encoding approaches.
- Improved join search quality through our value-based post-processing using MinHash sketches and LSH Ensemble.
- Robustness across different table sampling techniques and search strategies, offering flexible performance-speed trade-offs.

Finally, our analysis of the used benchmarks highlights a high heterogeneity in benchmark characteristics such as the predominance of organizational entities in unionable search datasets versus the prevalence of numerical and temporal data in joinable datasets. This makes it difficult to interpret and fairly compare the performance results and generalizability of the different methods.

Conclusion Our work shows that graph-based TaLMs like HyTrel can effectively support data discovery while better preserving tabular data properties. Our benchmark analysis uncovered patterns suggesting that evaluation outcomes may be influenced not only by model architectures but also by the inherent characteristics of the benchmarks themselves, warranting further investigation into potential evaluation biases. This ties into the broader paradigm shift towards foundation database models [13], where generalizable pre-trained models like HyTrel can be leveraged to avoid task-specific architectures. Future work includes exploring fine-tuning strategies, investigating scalability optimizations for larger data lakes [2], and studying the correlation between benchmark characteristics and model performance.

References

1. P. Chen, S. Sarkar, L. Lausen, B. Srinivasan, S. Zha, R. Huang, and G. Karypis. Hytrel: Hypergraph-enhanced tabular data representation learning. *Advances in Neural Information Processing Systems*, 36, 2024.
2. Y. Deng, C. Chai, L. Cao, Q. Yuan, S. Chen, Y. Yu, Z. Sun, J. Wang, J. Li, Z. Cao, K. Jin, C. Zhang, Y. Jiang, Y. Zhang, Y. Wang, Y. Yuan, G. Wang, and N. Tang. Lakebench: A benchmark for discovering joinable and unionable tables in data lakes. *Proceedings of the VLDB Endowment*, 17(8):1925–1938, 2024.
3. Y. Dong, C. Xiao, T. Nozawa, M. Enomoto, and M. Oyamada. DeepJoin: Joinable Table Discovery with Pre-Trained Language Models. *Proceedings of the VLDB Endowment*, 16(10):2458–2470, June 2023.
4. M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. The faiss library. 2024.
5. G. Fan, J. Wang, Y. Li, D. Zhang, and R. Miller. Starmie: Semantics-aware dataset discovery from data lakes with contextualized column-based representation learning. *arXiv preprint arXiv:2210.01922*, 2023.
6. X. Fang, W. Xu, F. A. Tan, Z. Hu, J. Zhang, Y. Qi, S. H. Sengamedu, and C. Faloutsos. Large language models (llms) on tabular data: Prediction, generation, and understanding - A survey. *Trans. Mach. Learn. Res.*, 2024, 2024.
7. J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.
8. A. Khatiwada, G. Fan, R. Shraga, Z. Chen, W. Gatterbauer, R. J. Miller, and M. Riedewald. Santos: Relationship-based semantic table union search. *Proceedings of the ACM on Management of Data*, 1(1):1–25, 2023.
9. A. Khatiwada, H. Kokel, I. Abdelaziz, S. Chaudhury, J. Dolby, O. Hassanzadeh, Z. Huang, T. Pedapati, H. Samulowitz, and K. Srinivas. Tabsketchfm: Sketch-based tabular representation learning for data discovery over data lakes. *IEEE ICDE*, 2025.
10. C.-T. Li, Y.-C. Tsai, and J. C. Liao. Graph neural networks for tabular data learning. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 3589–3592. IEEE, 2023.
11. F. Nargesian, E. Zhu, K. Q. Pu, and R. J. Miller. Table union search on open data. *Proceedings of the VLDB Endowment*, 11(7):813–825, 2018.
12. K. Srinivas, J. Dolby, I. Abdelaziz, O. Hassanzadeh, H. Kokel, A. Khatiwada, T. Pedapati, S. Chaudhury, and H. Samulowitz. Lakebench: Benchmarks for data discovery over data lakes. *arXiv preprint arXiv:2307.04217*, 2023.
13. J. Wehrstein, C. Binnig, F. Özcan, S. Vasudevan, Y. Gan, and Y. Wang. Towards foundation database models.
14. E. Zhu, F. Nargesian, K. Q. Pu, and R. J. Miller. Lsh ensemble: Internet-scale domain search. *arXiv preprint arXiv:1603.07410*, 2016.