Something's Fishy In The Data Lake: A Critical Re-evaluation of Table Union Search Benchmarks

Allaa Boutaleb, Bernd Amann, Hubert Naacke and Rafael Angarita

Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
{firstname.lastname}@lip6.fr

Abstract

Recent table representation learning and data discovery methods tackle table union search (TUS) within data lakes, which involves identifying tables that can be unioned with a given query table to enrich its content. These methods are commonly evaluated using benchmarks that aim to assess semantic understanding in real-world TUS tasks. However, our analysis of prominent TUS benchmarks reveals several limitations that allow simple baselines to perform surprisingly well, often outperforming more sophisticated approaches. This suggests that current benchmark scores are heavily influenced by dataset-specific characteristics and fail to effectively isolate the gains from semantic understanding. To address this, we propose essential criteria for future benchmarks to enable a more realistic and reliable evaluation of progress in semantic table union search.

1 Introduction

Measurement enables scientific progress. In computer science and machine learning, this requires the creation of efficient benchmarks that provide a stable foundation for evaluation, ensuring that observed performance scores reflect genuine capabilities for real-world tasks.

Table Union Search (TUS) aims to retrieve tables C from a corpus that are semantically unionable with a query table Q, meaning they represent the same information type and permit vertical concatenation (row appending) (Nargesian et al., 2018; Fan et al., 2023a). As a top-k retrieval task, TUS ranks candidate tables C by a table-level relevance score $R(Q, C)$. This score is typically obtained by aggregating column-level semantic relevance scores $R(C_Q, C_C)$ computed for each column C_Q of the query table Q and each column C_C of the candidate table C. The aggregation often involves finding an optimal mapping between the columns of Q and C, for instance via maximum bipartite matching (Fan et al., 2023b). Successful TUS facilitates data integration and dataset enrichment (Khatiwada et al., 2023; Castelo et al., 2021).

Recent research has introduced sophisticated TUS methods with complex representation learning (Fan et al., 2023b; Khatiwada et al., 2025; Chen et al., 2023) designed to capture deeper semantics. However, current benchmarks often exhibit excessive schema overlap, limited semantic complexity, and potential ground truth inconsistencies, which raises questions about whether they provide a reliable environment to evaluate advanced TUS capabilities. While state-of-the-art methodologies leverage semantic reasoning to reflect the task specific challenges, observed high performance may be significantly attributed to model adaptation to specific statistical and structural properties inherent within the benchmark datasets. This phenomenon can confound the accurate assessment and potentially underestimate the isolated contribution of improvements specifically targeting semantics-aware TUS.

In this paper, we examine prominent TUS benchmarks [1], using simple baselines to assess the benchmarks themselves. Our research questions are:

1. Do current TUS benchmarks necessitate deep semantic analysis, or can simpler features achieve competitive performance?
2. How do benchmark properties and ground truth quality impact TUS evaluation?
3. What constitutes a more realistic and discriminative TUS benchmark?

Our analysis[2] reveals that simple baseline methods often achieve surprisingly strong performance by leveraging benchmark characteristics rather than demonstrating sophisticated semantic reasoning.

[1] Preprocessed benchmarks used in our evaluation are available at https://zenodo.org/records/15499092

[2] Our code is available at: https://github.com/Allaa-boutaleb/fishy-tus

Proceedings of the 4th Table Representation Learning Workshop, pages 71–85

Our contributions include:

- A systematic analysis identifying limitations in current TUS benchmarks.
- Empirical evidence showing simple embedding methods achieve competitive performance.
- An investigation of ground truth reliability issues across multiple TUS benchmarks.
- Criteria for developing more realistic and discriminative benchmarks.

2 Related Work

We review existing research on TUS methods and the benchmarks used for their evaluation, with a focus on how underlying assumptions about table unionability have evolved to become increasingly nuanced and complex.

2.1 Methods and Their Assumptions

2.1.a) Foundational Approaches: Following early work on schema matching and structural similarity (Sarma et al., 2012), Nargesian et al. (2018) formalized TUS by assessing attribute unionability via value overlap, ontology mappings, and natural language embeddings. Bogatu et al. (2020) incorporated additional features (e.g., value formats, numerical distributions) and proposed a distinct aggregation method based on weighted feature distances. Efficient implementations of these methods rely on Locality Sensitive Hashing (LSH) indices and techniques like LSH Ensemble (Zhu et al., 2016) for efficient table search.

2.1.b) Incorporating Column Relationships: Beyond considering columns individually, Khatiwada et al. (2023) proposed SANTOS, which evaluates the consistency of inter-column semantic relationships (derived using an existing knowledge base like YAGO (Pellissier Tanon et al., 2020) or by synthesizing one from the data itself) across tables to improve TUS accuracy.

2.1.c) Deep Table Representation Learning: Recent approaches use deep learning for tabular understanding. Pylon (Cong et al., 2023) and Starmie (Fan et al., 2023b) use contrastive learning for contextualized column embeddings. Hu et al. (2023) propose AutoTUS, employing multi-stage self-supervised learning. TabSketchFM (Khatiwada et al., 2025) uses data sketches to preserve semantics while enabling scalability. Graph-based approaches like HEARTS (Boutaleb et al., 2025) leverage HyTrel (Chen et al., 2023), representing tables as hypergraphs to preserve structural properties.

2.2 Benchmarks and their Characteristics

Benchmark creators make design choices at every stage of the construction process that reflect their understanding and assumptions about how and when tables can and should be meaningfully combined. We identify three primary construction paradigms applied for building TUS benchmarks:

2.2.a) Partitioning-based: $\text{TUS}_{\text{Small}}$ and $\text{TUS}_{\text{Large}}$ (Nargesian et al., 2018), as well as the SANTOS benchmark (referring to $\text{SANTOS}_{\text{Small}}$, as $\text{SANTOS}_{\text{Large}}$ is not fully labeled) (Khatiwada et al., 2023) partition seed tables horizontally or vertically, labeling tables from the same original seed as unionable with the seed table. This approach likely introduces significant schema and value overlap, potentially favoring methods that detect surface-level similarity rather than deeper semantic alignment.

2.2.b) Corpus-derived: The PYLON benchmark (Cong et al., 2023) curates tables from GitTables (Hulsebos et al., 2023) on specific topics. While this avoids systematic partitioning overlap, the focus on common topics may result in datasets with a general vocabulary that is well-represented in pre-trained models. This can reduce the comparative advantage of specialized table representation learning and data discovery methods.

2.2.c) LLM-generated: UGEN (Pal et al., 2024) leverages Large Language Models (LLMs) to generate table pairs, aiming to overcome limitations of previous methods by crafting purposefully challenging scenarios, including hard negatives. However, this strategy introduces the risk of ground truth inconsistency, as LLMs may interpret the criteria for unionability differently across generations, affecting label reliability.

2.2.d) Hybrid approaches: LAKEBENCH (Deng et al., 2024) uses tables from OpenData[3] and WebTable corpora[4] alongside both partitioning-based synthetic queries and real queries sampled from the corpus. However, such hybrid approaches can inherit the limitations of their constituent methods: partitioning still risks high overlap, candidate-based labeling may yield incomplete ground truth,

[3] https://data.gov/

[4] https://webdatacommons.org/webtables/

Benchmark		Overall Statistics					Column Type (%)				Size (MB)
		Files	Rows	Cols	Avg Shape	Missing%	Str	Int	Float	Other	
SANTOS	NQ	500	2,736,673	5,707	5473 × 11	9.96	65.39	17.00	11.46	6.15	*∼422*
	Q	50	1,070,085	615	21402 × 12	5.79	73.17	15.93	8.46	2.44	
$\text{TUS}_{\text{Small}}$	NQ	1,401	5,293,327	13,196	3778 × 9	6.77	85.43	5.93	4.77	3.86	*∼1162*
	Q	125	577,900	1,610	4623 × 13	6.86	82.05	7.08	5.84	5.03	
$\text{TUS}_{\text{Large}}$	NQ	4,944	8,416,415	53,133	1702 × 11	12.53	90.12	5.10	3.57	1.21	*∼1459*
	Q	100	213,229	1,792	2132 × 18	14.87	90.46	3.68	4.13	1.73	
PYLON	NQ	1,622	85,282	16,802	53 × 10	0.00	58.74	25.36	15.90	0.00	*∼22*
	Q	124	11,207	880	90 × 7	0.00	75.68	22.95	1.36	0.00	
UGEN_{V1}	NQ	1,000	7,609	10,315	8 × 10	5.79	91.68	3.27	4.29	0.76	*∼4*
	Q	50	405	546	8 × 11	5.87	90.48	4.58	4.21	0.73	
UGEN_{V2}	NQ	1,000	18,738	13,360	19 × 13	8.16	82.40	11.71	5.50	0.39	*∼8*
	Q	50	5,363	665	107 × 13	4.14	84.96	10.23	2.41	2.41	
LB-OpenData	NQ	4,832	351,067,113	89,757	72655 × 19	3.44	52.50	22.56	22.37	2.57	*∼80834*
	Q	3,138	238,576,481	61,815	76028 × 20	2.90	40.60	26.28	27.60	5.53	
LB-Webtable	NQ	29,686	1,039,347	387,432	35 × 13	0.01	61.07	26.28	12.64	0.01	*∼170*
	Q	5,488	335,187	56,174	61 × 10	0.00	40.43	43.06	16.51	0.01	

Table 1: Table Union Search Benchmarks Summary. NQ = Non-query table, Q = Query table.

and the large scale of these benchmarks can introduce practical evaluation challenges.

3 Methodology

As TUS methods become increasingly sophisticated, the benchmarks used for their evaluation may contain inherent characteristics that hinder the accurate assessment of progress in semantic understanding. This section outlines our approach to examining prominent TUS benchmarks through analysis of their construction methods and strategic use of simple baselines as diagnostic tools. The goal of advanced TUS methods is to capture deep semantic compatibility between tables, beyond simple lexical or structural similarity. Our investigation first analyzes the various benchmark construction processes to identify potential structural weaknesses, then employs computationally inexpensive baseline methods to reveal how these characteristics enable alternative pathways to high performance, thereby influencing evaluation outcomes.

3.1 Analyzing Benchmark Construction

We examine five prominent families of TUS benchmarks and formulate hypotheses about their potential limitations based on their construction methodologies (Table 1). We identify three issues stemming from these methodologies: **(1) excessive overlap**, **(2) semantic simplicity**, and **(3) ground truth inconsistencies**, which we detail below:

3.1.a) Excessive Overlap: Benchmarks like $\text{TUS}_{\text{Small}}$, $\text{TUS}_{\text{Large}}$, SANTOS, and the *synthetic* query portion of the LAKEBENCH derivatives are created by partitioning seed tables horizontally and vertically, with tables derived from the same original seed designated as unionable pairs. We hypothesize that this methodology inherently leads to significant overlap in both schema (column names) and content (data values) between query tables and their ground truth unionable candidates.

To quantify this, we measure overlap using the Szymkiewicz–Simpson coefficient for exact column names ($Overlap_c$, Eq. 1) and for values of a given data type d ($Overlap_v$, Eq. 2) between ground truth pairs.

$$Overlap_c(Q, C) = \frac{|Cols_Q \cap Cols_C|}{\min(|Cols_Q|, |Cols_C|)} \quad (1)$$

$$Overlap_v(Q, C) = \frac{|V_Q^d \cap V_C^d|}{\min(|V_Q^d|, |V_C^d|)} \quad (2)$$

where $Cols_Q$ and $Cols_C$ denote the sets of column names in the query table Q and candidate table C respectively, and V_Q^d, V_C^d represent the sets of unique values of data type d in each table. The coefficient equals 1.0 when one set is fully contained within the other. Figure 1 shows the distribution of overlap coefficients, with values $\geq 50\%$ indicating substantial overlap. As expected, partitioning-based benchmarks exhibit high overlap: over 90% of ground truth pairs share $\geq 50\%$ of exact column names. For value overlap, we focus on string data types, which dominate the benchmarks (Table 1). Here too, 45–60% of query-candidate pairs share $\geq 50\%$ of string tokens. LAKEBENCH derivatives (LB-OPENDATA, LB-WEBTABLE) show similar trends. Appendix A provides a detailed breakdown by data type. This high surface similarity favors simple lexical methods

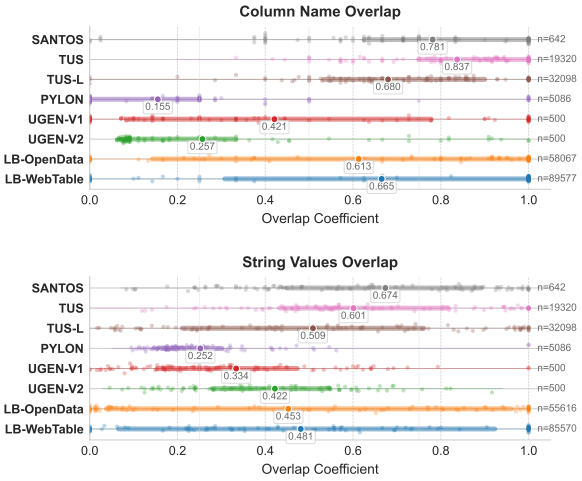

Figure 1: Distribution of Exact Column Name Overlap (Top) and String Value Overlap (Bottom) Coefficients for Ground Truth Unionable Pairs Across Benchmarks. Colored circles represent mean values; numbers on the right indicate total pairwise relationships considered.

and also influences advanced models by introducing repeated patterns in serialized inputs (Starmie), data sketches (TabSketchFM), and graph structures (HEARTS). Though designed for deeper semantics, these models are affected by strong benchmark-induced surface signals, making it hard to attribute performance gains purely to nuanced reasoning.

3.1.b) Semantic Simplicity: Benchmarks derived directly from large corpora, such as PYLON (Cong et al., 2023) using GitTables (Hulsebos et al., 2023) or the *real* query portions of LAKEBENCH derivatives using diverse public datasets, avoid the systematic overlap introduced by partitioning. However, we hypothesize that this construction method introduces other limitations since (1) it often focuses on relatively common topics with simpler semantics, reducing the need for specialized domain knowledge, and (2) it generally draws from public data sources likely included in the pre-training corpora of large foundation models. Evidence from specific benchmarks supports this concern. PYLON's construction indeed avoids high overlap (Figure 1 shows lower overlap than partitioning-based benchmarks). For LAKEBENCH, while the distinction between *real* and *synthetic* queries was unavailable during our analysis[5], the significant overall observed overlap suggests that synthetic, partitioning-based queries constitute a large portion of the benchmark. The semantic simplicity evident in PYLON's topics and the public origins of data in both PYLON and LAKEBENCH could favor general-purpose models like BERT (Devlin et al., 2019) or SBERT (Reimers and Gurevych, 2019), which have with a high, however unverifiable, probability encountered similar content during pre-training. Consequently, the semantic challenge presented by these benchmarks might be relatively low for models with strong general language understanding – a contrast to documented LLM struggles with non-public, enterprise-specific data characteristics (Bodensohn et al., 2025), potentially allowing off-the-shelf embedding models to achieve high performance without fine-tuning.

3.1.c) Noisy Ground Truths: Ensuring accurate and complete ground truth labels is challenging, especially with automated generation or large-scale human labeling efforts as used in LLM-generated benchmarks (UGEN) and large human-labeled ones (LAKEBENCH derivatives). We hypothesize that ground truth in these benchmarks may suffer from reliability issues, including incorrect labels (false positives/negatives) or incompleteness (missed true positives). For UGEN, generating consistent, accurate positive and negative pairs (especially hard negatives) is difficult. LLMs might interpret unionability rules inconsistently across generations, leading to noisy labels. For large-scale human labeling with LB-OPENDATA and LB-WEBTABLE, the process introduces two risks: *incompleteness*, if the initial retrieval misses true unionable tables; and *incorrectness*, if human judgments vary or contain errors despite validation efforts. Evaluating performance on UGEN and LAKEBENCH derivatives thus requires caution. Scores are affected by label noise or incompleteness; low scores reflect ground truth issues and are therefore not solely attributable to benchmark difficulty, while the maximum achievable recall is capped by unlabeled true positives.

3.2 Baseline Methods for Benchmark Analysis

Based on the hypothesized benchmark issues identified above, we select some simple baseline methods to test benchmark sensitivity to different information types. While the (1) overlap and (2) general semantics limitations can be directly examined through baseline performance, (3) the ground truth integrity issue requires separate validation of labels, which we address in Section 5.2. Detailed implementation choices for all baseline methods are in Appendix B.1.

[5] https://github.com/RLGen/LakeBench/issues/9

3.2.a) Bag-of-Words Vectorizers: To test whether the *Excessive Overlap* enables methods sensitive to token frequency to perform well on partitioning-based benchmarks, we employ standard lexical vectorizers (`HashingVectorizer`, `TfidfVectorizer`, and `CountVectorizer`) from scikit-learn[6]. These generate column embeddings based on sampled string values, with a single table vector obtained via *max pooling* across column vectors. These baselines test whether high performance can be achieved primarily by exploiting surface signals without semantic reasoning.

3.2.b) Pre-trained Sentence Transformers: To examine whether the *Semantic Simplicity* allows benchmarks from broad corpora to be effectively processed by pre-trained language models, we use a Sentence-BERT model (`all-mpnet-base-v2`[7]) with three column-to-text serializations: (1) SBERT (V+C): input includes column name and sampled values; (2) SBERT (C): input is only the column name; and (3) SBERT (V): input is only concatenated sampled values. Column embeddings are aggregated using mean pooling to produce a single table vector. These baselines assess whether general semantic embeddings, without task-specific fine-tuning, suffice for high performance on benchmarks with general vocabulary.

4 Experimental Setup

To evaluate our hypotheses about benchmark limitations, we employ both simple baseline methods (Section 3.2) and advanced SOTA methods in a controlled experimental framework. This section details the benchmark datasets used, any necessary preprocessing, the comparative methods, and our standardized evaluation approach.

4.1 Benchmarks

Our analysis uses the benchmarks described in Section 2.2, with post-preprocessing statistics summarized in Table 1. Most benchmarks were used as-is, but the large-scale LAKEBENCH derivatives (LB-OPENDATA and LB-WEBTABLE) required additional preprocessing for feasibility and reproducibility. The original datasets were too large to process directly and included practical issues, such as missing files, as well as characteristics that complicated evaluation, such as many unreferenced tables. We removed ground truth entries pointing to missing files (58 in LB-WEBTABLE), and excluded unreferenced tables from the retrieval corpus (removing $\sim$5,300 and $>$2.7M files from LB-OPENDATA and LB-WEBTABLE, respectively). This latter step was done purely for computational feasibility; as a side effect, it simplifies the benchmark by eliminating tables that would otherwise be false positives if retrieved. We also ensured that each query table was listed as a candidate for itself. These steps substantially reduced corpus size while preserving evaluation integrity. The LAKEBENCH variants considered in our study are those available as of May 20, 2025[8]. Future updates to the original repository may modify dataset contents, which yield different evaluation results.

Additionally, for LB-OPENDATA, we created a smaller variant with tables truncated to 1,000 rows, which we use in experiments alongside the original version (Table 2). For $\text{TUS}_{\text{Small}}$ and $\text{TUS}_{\text{Large}}$, we followed prior work (Fan et al., 2023b; Hu et al., 2023), sampling 125 and 100 queries, respectively. For the other benchmarks, all queries were used.

4.2 Comparative Methods

To evaluate our baseline methods (Section 3.2), we compare them against key TUS models previously discussed in Section 2.1, focusing on SOTA methods. For each method, we optimize implementation using publicly available code for fairness:

- **Starmie** (Fan et al., 2023b): We retrained the RoBERTa-based model for 10 epochs on each benchmark using recommended hyperparameters and their "Pruning" bipartite matching search strategy for generating rankings, which achieves optimal results according to the original paper.
- **HEARTS** (Boutaleb et al., 2025): We utilized pre-trained HyTrel embeddings (Chen et al., 2023) with a contrastively-trained checkpoint. For each benchmark, we adopted the best-performing search strategy from the HEARTS repository: Cluster Search for SANTOS, PYLON, and UGEN benchmarks, and ANN index search with max pooling for the TUS and LAKEBENCH benchmarks.
- **TabSketchFM** (Khatiwada et al., 2025): Results for the $\text{TUS}_{\text{Small}}$ and SANTOS were reported directly from the original paper, as the pretrained checkpoint was unavailable at the time of our experiments.

[6] Scikit-Learn Vectorizers Documentation
[7] all-mpnet-base-v2 on Hugging Face
[8] LakeBench commit df7559d used in our study

These methods represent significant advancements in table representation learning. AutoTUS (Hu et al., 2023) wasn't included due to code unavailability at the time of writing. We provide further implementation details in Appendix B.2.

4.3 Evaluation Procedure

We use a consistent evaluation procedure for all baseline and SOTA methods to ensure fair comparison. Table vectors are generated per method (Section 3.2 for baselines; SOTA-specific procedures otherwise) and L2-normalized for similarity via inner product. For similarity search, baseline methods use the FAISS library (Douze et al., 2024) with an exact inner product index (`IndexFlatIP`); each query ranks all candidate tables by similarity. SOTA methods use FAISS or alternative search strategies (Appendix B.2). Following prior work (Fan et al., 2023b; Hu et al., 2023), we report Precision@k (P@k) and Recall@k (R@k), averaged across queries. Values of k follow prior works and are shown in results tables (e.g., Table 2). We also evaluate computational efficiency via offline (training, vector extraction, indexing) and online (query search) runtimes, with hardware details in Appendix B.3.

5 Results and Discussion

Our empirical evaluation revealed significant patterns across benchmarks that expose fundamental limitations in their ability to measure progress in semantic understanding. Tables 2 and 3 present effectiveness and efficiency metrics respectively.

5.1 Evidence of Benchmark Limitations

The most compelling evidence for our benchmark limitation hypotheses emerges from the unexpectedly strong performance of simple baselines. On partitioning-based benchmarks (TUS$_{\text{Small}}$, TUS$_{\text{Large}}$, SANTOS), lexical methods achieve near-perfect precision, matching or exceeding sophisticated models at a fraction of the cost. This directly validates our overlap issue hypothesis: the high schema and value overlap (Figure 1) creates trivial signals that simple lexical matching can exploit. While advanced methods like Starmie or HEARTS also achieve high scores here, the fact that much simpler, non-semantic methods perform nearly identically leads us to conclude that the benchmark itself does not effectively differentiate methods based on deep semantic understanding. This phenomenon, where simpler approaches can achieve comparable or even better results than more complex counterparts, especially when computational costs are considered, has also been observed in related data lake tasks such as table augmentation via join search (Cappuzzo et al., 2024).

For PYLON, a different pattern emerges: lexical methods perform considerably worse due to the much lower exact overlap, but general-purpose semantic embeddings excel. SBERT variants, particularly SBERT(V+C) combining column and value information, outperform specialized SOTA models like Starmie. This confirms our general semantics hypothesis that these benchmarks employ vocabulary well-represented in standard pre-trained embeddings, diminishing the advantage of specialized tabular architectures for the TUS task.

LB-OPENDATA and LB-WEBTABLE exhibit both limitations despite their scale. Simple lexical methods remain surprisingly competitive, while SBERT variants consistently outperform specialized models. The computational demands of sophisticated models create additional practical barriers: Starmie requires substantial offline costs (training and inference) plus over 16 hours to process the queries on the truncated LB-OPENDATA, and over 21 hours to evaluate the queries of LB-WEBTABLE. HEARTS performs better computationally by leveraging a pre-trained checkpoint without additional training, resulting in a shorter offline processing time, but still under-performs SBERT variants.

5.2 Ground Truth Reliability Issues

A notable observation across UGEN and LAKEBENCH derivatives is the significant gap between the R@k achieved by all methods and the IDEAL recall (Table 2). This discrepancy led us to question the reliability of the benchmarks' ground truth labels. We hypothesized that such gaps might indicate not only the limitations of the search methods or the inherent difficulty of the benchmarks but also potential incompleteness or inaccuracies within the ground truth itself. Examining discrepancies at small values of k is particularly revealing, as this scrutinizes the highest-confidence predictions of a system. If a high-performing method frequently disagrees with the ground truth at these top ranks, it may signal issues with the ground truth labels.

To investigate this, we defined two heuristic metrics designed to help identify potential ground truth flaws. Let $\mathcal{Q} = \{Q_1, \ldots, Q_N\}$ be N query tables. For $Q_i \in \mathcal{Q}$, $C_{Q_i,k}$ is the set of top-k candidates

Method	SANTOS		TUS		$\text{TUS}_{\text{Large}}$		PYLON		UGEN_{V1}		UGEN_{V2}		LB-OpenData$_{1k}$		LB-OpenData		LB-WebTable	
	P@10	R@10	P@60	R@60	P@60	R@60	P@10	R@10	P@10	R@10	P@10	R@10	P@50	R@50	P@50	R@50	P@20	R@20
IDEAL	1.00	0.75	1.00	0.34	1.00	0.23	1.00	0.24	1.00	1.00	1.00	1.00	0.39	1.00	0.39	1.00	0.81	0.95
Non-specialized Embedding Methods																		
HASH	0.98	0.74	0.99	0.33	**0.99**	**0.23**	0.64	0.15	0.59	0.59	0.43	0.43	0.21	0.60	0.21	0.60	0.21	0.25
TFIDF	**0.99**	**0.74**	**1.00**	**0.34**	**0.99**	**0.23**	0.70	0.17	0.58	0.58	0.50	0.50	0.21	0.61	0.21	0.61	0.23	0.27
COUNT	**0.99**	**0.74**	**1.00**	**0.34**	**0.99**	**0.23**	0.68	0.17	0.58	0.58	0.50	0.50	0.21	0.60	0.21	0.60	0.23	0.27
SBERT (V+C)	0.98	0.74	**1.00**	**0.34**	**0.99**	**0.23**	**0.91**	**0.22**	**0.61**	**0.61**	**0.68**	**0.68**	**0.23**	**0.66**	**0.23**	**0.66**	**0.26**	**0.31**
SBERT (V)	0.94	0.71	**1.00**	**0.34**	**0.99**	**0.23**	0.84	0.20	0.58	0.58	0.58	0.58	0.22	0.62	0.22	0.62	0.25	0.29
SBERT (C)	0.98	0.74	**1.00**	**0.34**	0.98	0.23	0.85	0.21	0.60	0.60	0.65	0.65	0.22	0.64	0.22	0.64	0.16	0.20
Specialized Table Union Search Methods																		
Starmie	0.98	0.73	0.96	0.31	0.93	0.21	0.81	0.20	0.57	0.57	0.58	0.58	0.18	0.51	‡	‡	0.25	0.30
HEARTS	0.98	0.74	**1.00**	**0.34**	**0.99**	**0.23**	0.65	0.16	0.56	0.56	0.37	0.37	0.19	0.61	0.19	0.60	0.23	0.28
TabSketchFM	0.92	0.69	0.97	0.32	*	*	*	*	*	*	*	*	*	*	*	*	*	*

Table 2: Precision and Recall across benchmarks. Highest values in **bold**, second highest underlined. IDEAL represents the maximum possible P@k and R@k achievable for each benchmark at the specified k. ∗ : Results unavailable as checkpoint was not publicly accessible. ‡ : Not reported due to excessive computational requirements.

Method	SANTOS		TUS		$\text{TUS}_{\text{Large}}$		PYLON		UGEN_{V1}		UGEN_{V2}		LB-OpenData$_{1k}$		LB-OpenData		LB-WebTable	
	Offline	Online	Offline	Online	Offline	Online	Offline	Online	Offline	Online	Offline	Online	Offline	Online	Offline	Online	Offline	Online
Non-specialized Embedding Methods																		
HASH	0m 15s	0m 0s	0m 43s	0m 1s	1m 45s	0m 2s	0m 19s	0m 1s	0m 12s	0m 0s	0m 14s	0m 0s	7m 56s	0m 31s	12m 4s	0m 22s	6m 3s	0m 21s
TFIDF/COUNT	0m 53s	0m 0s	1m 45s	0m 1s	3m 10s	0m 2s	0m 22s	0m 1s	0m 9s	0m 0s	0m 12s	0m 0s	22m 22s	0m 31s	37m 14s	0m 21s	6m 21s	0m 22s
SBERT	1m 45s	0m 0s	3m 30s	0m 0s	9m 21s	0m 15s	3m 18s	0m 0s	1m 41s	0m 0s	2m 20s	0m 0s	27m 47s	0m 4s	82m 13s	0m 4s	30m 45s	0m 3s
Specialized Table Union Search Methods																		
STARMIE	19m 3s	1m 2s	4m 24s	8m 59s	14m 43s	20m 29s	7m 56s	3m 27s	2m 8s	1m 0s	2m 45s	1m 45s	131m 48s	1220m 53s	–	–	48m 11s	1311m 43s
HEARTS	0m 21s	0m 34s	1m 1s	0m 0s	3m 10s	0m 0s	0m 57s	0m 36s	0m 23s	0m 40s	0m 30s	0m 35s	21m 33s	0m 3s	76m 12s	0m 5s	29m 28s	0m 3s

Table 3: Computational efficiency across benchmarks. Times are averaged over 5 runs due to runtime variability. Offline includes vector generation, indexing, and training times where applicable; Online is total query search time.

retrieved by a search method for Q_i, and G_{Q_i} is the set of ground truth candidates labeled unionable with Q_i.

1. **GTFP@k (Ground Truth False Positive Rate)**: This measures the fraction of top-k candidates retrieved by a search method that are not labeled as unionable in the original ground truth. A high GTFP@k, especially at small k, suggests the method might be identifying valid unionable tables missing from the ground truth, thereby helping us pinpoint its possible *incompleteness*. It is calculated as:

$$\frac{\sum_{i=1}^{N} |C_{Q_i,k} \setminus G_{Q_i}|}{N \cdot k}$$

Here, $|C_{Q_i,k} \setminus G_{Q_i}|$ counts retrieved candidates for Q_i that are absent from its ground truth set G_{Q_i}. The denominator is the total top-k slots considered across all queries.

2. **GTFN@k (Ground Truth False Negative Rate)**: This quantifies the fraction of items labeled as positives in the ground truth that a well-performing search method fails to retrieve within its top-k results (considering a capped expectation up to k items per query). It is calculated as:

$$\frac{\sum_{i=1}^{N} (\min(k, |G_{Q_i}|) - |G_{Q_i} \cap C_{Q_i,k}|)}{\sum_{i=1}^{N} \min(k, |G_{Q_i}|)}$$

The term $\min(k, |G_{Q_i}|)$ represents the capped ideal number of ground truth items we would expect to find in the top k for Q_i. The numerator sums the "misses" for each query: the difference between this capped ideal and the number of ground truth items actually retrieved. The denominator sums this capped ideal across all queries. A high GTFN@k at small k is particularly insightful when investigating ground truth integrity. If we trust the method's ability to discern relevance, a high GTFN@k implies that the method correctly deprioritizes items that, despite being in the ground truth, might be less relevant or even incorrectly labeled as positive. Thus, it can signal potential *incorrectness* within the ground truth. GTFN@k is equivalent to "1 − CappedRecall@k" (Thakur et al., 2021).

These metrics assume discrepancies between a strong search method and the ground truth may indicate flaws in the latter. While not highly accurate, they helped us identify a smaller, focused subset of query-candidate pairs with disagreements for deeper manual or LLM-based inspection. Results are shown in Table 4.

Beyond heuristic metrics, we also conduct a more direct–though still imperfect–assessment of UGEN's ground truth using an LLM-as-a-judge approach. While this method may not capture the same conflicts identified by the cheaper GTFP/GTFN heuristics, it provides a complementary perspective that can offer more precise insights in certain cases. We use

`gemini-2.0-flash-thinking-exp-01-21`[9], chosen for its 1M-token context window, baked-in reasoning abilities, and low hallucination rate[10]. This LLM-as-a-judge approach has become increasingly common in recent works (Gu et al., 2024; Wolff and Hulsebos, 2025). We gave the LLM both tables in each query-candidate pair, along with a detailed prompt including curated unionable and non-unionable examples from UGEN (see Appendix D) to condition the LLM's understanding of unionability based on the benchmark. Each pair was evaluated in 5 independent runs with `temperature=0.1`. A sample of 20 LLM outputs was manually validated and showed strong alignment with human judgment. Comparison with original UGEN labels (Table 5) revealed substantial inconsistencies. Our manual inspection (Appendix C.1) suggested the LLM often provided more accurate assessments, indicating notable noise in the original ground truth.

Given the scale of LB-OPENDATA and LB-WEBTABLE, full LLM adjudication was impractical. Instead, we used SBERT(V+C) as our reference search method to compute GTFP@k, focusing on top-ranked pairs not labeled as unionable in the ground truth. As shown in Table 4, such cases were frequent even at top ranks ($2 < k < 5$). To assess ground truth completeness, we manually inspected 20 randomly sampled top-2 and top-3 disagreements. Of these, 19 were genuinely unionable but missing from the ground truth; the remaining pair was correctly non-unionable, with SBERT likely misled by its numeric-only columns. These results suggest non-negligible *incompleteness* in the LAKEBENCH ground truth. Example cases are shown in Appendix C.2.

In summary, our investigations, combining heuristic metrics, LLM-based adjudication, and manual inspection, reveal the presence of non-negligible noise and incompleteness within the original benchmark labels for both UGEN and LAKEBENCH. Consequently, performance metrics reported on these benchmarks may be influenced by these underlying ground truth issues, potentially misrepresenting true task difficulty or method capabilities.

5.3 Implications for Measuring Progress

Our experiments reveal several critical issues. Benchmark scores often fail to measure true semantic capabilities, as simple lexical or general embedding methods can match or outperform specialized models by exploiting excessive domain overlap, semantic simplicity, or ground truth inconsistency. This suggests that current benchmarks may inadvertently reward adaptation to these characteristics, making it difficult to quantify the practical benefits of progress on sophisticated TUS methods capabilities within these settings. These persistent issues also point to a fundamental challenge, the lack of a precise, operational definition for unionability, mirroring broader difficulties in dataset search (Hulsebos et al., 2024) and highlighting the need to address the subjective, context-dependent nature of table compatibility in practice.

[9] Gemini 2.0 Flash Thinking Model Card
[10] Vectara Hallucination Leaderboard

Benchmark (Metric)	@1	@2	@3	@4	@5
UGEN_{V1} (GTFP)	0.160	0.210	0.247	0.275	0.308
UGEN_{V1} (GTFN)	0.160	0.210	0.247	0.275	0.308
UGEN_{V2} (GTFP)	0.060	0.080	0.093	0.140	0.156
UGEN_{V2} (GTFN)	0.060	0.080	0.093	0.140	0.156
LB-OPENDATA (GTFP)	0.000	0.059	0.092	0.123	0.154
LB-OPENDATA (GTFN)	0.000	0.054	0.080	0.105	0.132
LB-WEBTABLE (GTFP)	0.000	0.110	0.198	0.296	0.377
LB-WEBTABLE (GTFN)	0.000	0.110	0.197	0.295	0.376

Table 4: Disagreement rates of top-k retrieved results between SBERT and the ground truth across different benchmarks. For UGEN, the query table is not considered a candidate to itself, so values at @1 reflect actual disagreement. For LAKEBENCH variants, the ground truth is normalized to include the query table as a valid candidate for itself. Therefore, the top-1 match is always correct by construction, yielding no disagreement @1.

GT Label	LLM Judge	UGEN V1	UGEN V2
Unionable	Non-unionable	24.8%	0.0%
Non-unionable	Unionable	33.8%	23.6%
Non-unionable	Non-unionable	16.2%	76.4%
Unionable	Non-unionable	25.2%	0.0%

Table 5: Breakdown of agreement and disagreement between ground truth labels and LLM-based judgments.

6 Towards Better TUS Benchmarks

In industry practice, unionability judgments are inherently subjective, depending on analytical goals, domain contexts, data accessibility constraints (Martorana et al., 2025), and user preferences (Mirzaei and Rafiei, 2023). Yet current benchmarks impose fixed definitions, creating a disconnect with practical utility: methods excelling on benchmarks often falter in real-world scenarios demanding different compatibility thresholds. Addressing this requires benchmark designs that embrace contextual variability and provide a stable foundation for

evaluation, lest even advanced methods fall short in practice.

Rethinking Benchmark Design Principles: Overcoming current benchmark limitations requires a shift in design focusing on three key principles: (1) actively reducing artifactual overlap while introducing controlled semantic heterogeneity to better reflect real-world schema and value divergence; (2) incorporating realistic domain complexity beyond general vocabularies, addressing challenges like non-descriptive schemas and proprietary terms where LLMs struggle (Bodensohn et al., 2025), thus emphasizing domain-specific training that may require industry collaboration; and (3) rethinking ground truth representation by replacing brittle binary labels with richer, nuanced formats validated through multi-stage adjudication to improve completeness and consistency.

Exploring Implementation Pathways: Translating these principles into practice requires concrete strategies for benchmark design and evaluation. One approach is to develop (1) scenario-driven micro-benchmarks targeting specific challenges such as schema drift simulation or value representation noise, enabling more granular analysis than coarse end-to-end metrics. Another is (2) advancing controllable synthetic data generation, following LLM-based methods like UGEN (Pal et al., 2024), to verifiably embed semantic constraints or domain knowledge, supporting diverse testbeds when real data is unavailable or sensitive. Equally important is (3) exploring adaptive, interactive evaluation frameworks such as human-in-the-loop systems, which would dynamically adjust relevance criteria based on user feedback to better capture the subjective nature of unionability. Tools like LakeVisage (Hu et al., 2025) further enhance usability and trust by recommending visualizations that help users interpret relationships among returned tables, improving transparency and interpretability in union search systems.Incorporating natural language preferences is also key. The recent NLCTABLES benchmark (Cui et al., 2025) advances this by introducing NL conditions for union and join searches on column values and table size constraints. However, its predicate-style conditions may be better addressed via post-retrieval filtering (e.g., translating NL to SQL predicates with an LLM), avoiding early discard of unionable candidates and unnecessary retrieval model complexity. To drive further advancement, benchmarks should incorporate (4) natural language conditions that capture key aspects of unionability and joinability, including specifications about the characteristics of the final integrated table or conditional integration logic. For example, a challenging predicate might require identifying tables that can be "joined with a query table on column A, unioned on columns B and C, and also contain an additional column D providing specific contextual information about a particular attribute." Such conditions would demand deeper reasoning capabilities from data integration systems and encourage the development of more sophisticated methods for Table Union and Join Search. Finally, moving beyond binary success metrics, future benchmarks could adopt (5) multi-faceted evaluation frameworks using richer ground truth representations to assess unionability across dimensions like schema compatibility, semantic type alignment, value distribution similarity, and task-specific relevance, offering a more holistic evaluation than current standards.

7 Conclusion

Our analysis of TUS benchmarks highlights three major limitations: excessive overlap in partitioning-based datasets, semantics easily captured by pre-trained embeddings, and non-negligible ground-truth inconsistencies. The first two allow simple baselines to rival sophisticated models with far lower computational cost, showing that high performance isn't necessarily tied to advanced semantic reasoning. The third undermines evaluation validity, as scores may reflect misalignment with flawed ground truth rather than actual benchmark difficulty. This gap between benchmark performance and true semantic capability suggests current evaluations often reward adaptation to benchmark-specific artifacts. To address this, we propose design principles that better reflect the complex, subjective nature of real-world table union search.

Limitations: Our study examined selected benchmarks and methods, with broader evaluation potentially revealing more insight. Our investigation of ground truth issues in UGEN and LAKEBENCH, while systematic, identifies certain patterns without exhaustive quantification.

Future Work: Developing benchmarks aligned with our proposed criteria represents the next step towards ensuring that measured progress translates to meaningful real-world utility.

References

Jan-Micha Bodensohn, Ulf Brackmann, Liane Vogel, Anupam Sanghi, and Carsten Binnig. 2025. Unveiling challenges for llms in enterprise data engineering. *Preprint*, arXiv:2504.10950.

Alex Bogatu, Alvaro A. A. Fernandes, Norman W. Paton, and Nikolaos Konstantinou. 2020. D3L: Dataset Discovery in Data Lakes. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 709–720. ArXiv:2011.10427 [cs].

Allaa Boutaleb, Alaa Almutawa, Bernd Amann, Rafael Angarita, and Hubert Naacke. 2025. HEARTS: Hypergraph-based related table search. In *ELLIS workshop on Representation Learning and Generative Models for Structured Data*.

Riccardo Cappuzzo, Gaël Varoquaux, Aimee Coelho, and Paolo Papotti. 2024. Retrieve, merge, predict: Augmenting tables with data lakes. *CoRR*, abs/2402.06282.

Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. 2021. Auctus: a dataset search engine for data discovery and augmentation. *Proceedings of the VLDB Endowment*, 14(12):2791–2794.

Pei Chen, Soumajyoti Sarkar, Leonard Lausen, Balasubramaniam Srinivasan, Sheng Zha, Ruihong Huang, and George Karypis. 2023. Hytrel: Hypergraph-enhanced tabular data representation learning. *Advances in Neural Information Processing Systems*, 36:32173–32193.

Tianji Cong, Fatemeh Nargesian, and H. V. Jagadish. 2023. Pylon: Semantic table union search in data lakes. *CoRR*, abs/2301.04901.

Lingxi Cui, Huan Li, Ke Chen, Lidan Shou, and Gang Chen. 2025. Nlctables: A dataset for marrying natural language conditions with table discovery. *CoRR*, abs/2504.15849.

Yuhao Deng, Chengliang Chai, Lei Cao, Qin Yuan, Siyuan Chen, Yanrui Yu, Zhaoze Sun, Junyi Wang, Jiajun Li, Ziqi Cao, Kaisen Jin, Chi Zhang, Yuqing Jiang, Yuanfang Zhang, Yuping Wang, Ye Yuan, Guoren Wang, and Nan Tang. 2024. LakeBench: A Benchmark for Discovering Joinable and Unionable Tables in Data Lakes. *Proceedings of the VLDB Endowment*, 17(8):1925–1938.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Grace Fan, Jin Wang, Yuliang Li, and Renée J. Miller. 2023a. Table Discovery in Data Lakes: State-of-the-art and Future Directions. In *Companion of the 2023 International Conference on Management of Data*, pages 69–75, Seattle WA USA. ACM.

Grace Fan, Jin Wang, Yuliang Li, Dan Zhang, and Renée J. Miller. 2023b. Semantics-aware dataset discovery from data lakes with contextualized column-based representation learning. *Proc. VLDB Endow.*, 16(7):1726–1739.

Daniel Gomm and Madelon Hulsebos. 2025. Metadata matters in dense table retrieval. In *ELLIS workshop on Representation Learning and Generative Models for Structured Data*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on llm-as-a-judge. *CoRR*, abs/2411.15594.

Xuming Hu, Shen Wang, Xiao Qin, Chuan Lei, Zhengyuan Shen, Christos Faloutsos, Asterios Katsifodimos, George Karypis, Lijie Wen, and Philip S. Yu. 2023. AUTOTUS: Automatic Table Union Search with Tabular Representation Learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3786–3800, Toronto, Canada. Association for Computational Linguistics.

Yihao Hu, Jin Wang, and Sajjadur Rahman. 2025. Lakevisage: Towards scalable, flexible and interactive visualization recommendation for data discovery over data lakes. *CoRR*, abs/2504.02150.

Madelon Hulsebos, Çagatay Demiralp, and Paul Groth. 2023. Gittables: A large-scale corpus of relational tables. *Proc. ACM Manag. Data*, 1(1):30:1–30:17.

Madelon Hulsebos, Wenjing Lin, Shreya Shankar, and Aditya Parameswaran. 2024. It Took Longer than I was Expecting: Why is Dataset Search Still so Hard? In *Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics*, pages 1–4, Santiago AA Chile. ACM.

Aamod Khatiwada, Grace Fan, Roee Shraga, Zixuan Chen, Wolfgang Gatterbauer, Renée J. Miller, and Mirek Riedewald. 2023. SANTOS: Relationship-based Semantic Table Union Search. *Proceedings of the ACM on Management of Data*, 1(1):1–25.

Aamod Khatiwada, Harsha Kokel, Ibrahim Abdelaziz, Subhajit Chaudhury, Julian Dolby, Oktie Hassanzadeh, Zhenhan Huang, Tejaswini Pedapati, Horst Samulowitz, and Kavitha Srinivas. 2025. Tabsketchfm: Sketch-based tabular representation learning for data discovery over data lakes. *IEEE ICDE*.

Margherita Martorana, Tobias Kuhn, and Jacco van Ossenbruggen. 2025. Metadata-driven table union search: Leveraging semantics for restricted access data integration. *CoRR*, abs/2502.20945.

Leland McInnes, John Healy, Steve Astels, and 1 others. 2017. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: uniform manifold approximation and projection. *J. Open Source Softw.*, 3(29):861.

Hamed Mirzaei and Davood Rafiei. 2023. Table union search with preferences. In *Joint Proceedings of Workshops at the 49th International Conference on Very Large Data Bases (VLDB 2023), Vancouver, Canada, August 28 - September 1, 2023*, volume 3462 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Fatemeh Nargesian, Erkang Zhu, Ken Q. Pu, and Renée J. Miller. 2018. TUS: Table union search on open data. *Proceedings of the VLDB Endowment*, 11(7):813–825.

Koyena Pal, Aamod Khatiwada, Roee Shraga, and Renée J Miller. 2024. Alt-gen: Benchmarking table union search using large language models. *Proceedings of the VLDB Endowment. ISSN*, 2150:8097.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. Yago 4: A reason-able knowledge base. In *The Semantic Web*, pages 583–596, Cham. Springer International Publishing.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Y Halevy, Hongrae Lee, Fei Wu, Reynold Xin, and Cong Yu. 2012. Finding related tables. In *SIGMOD Conference*, volume 10, pages 2213836–2213962.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Cornelius Wolff and Madelon Hulsebos. 2025. How well do llms reason over tabular data, really? *arXiv preprint arXiv:2505.07453*.

Erkang Zhu, Fatemeh Nargesian, Ken Q. Pu, and Renée J. Miller. 2016. LSH ensemble: Internet-scale domain search. *Proc. VLDB Endow.*, 9(12):1185–1196.

A Benchmark Overlap

As discussed in section 3.1.a), the degree of lexical overlap (both in column names and values) between query and candidate tables in benchmark ground truths can significantly influence model performance. Methods sensitive to surface-level similarity might perform well on benchmarks with high overlap without necessarily capturing deeper semantic relationships. This section provides a more detailed breakdown of overlap coefficients by data type across the different benchmarks evaluated. Figure 2 presents these distributions.

B Implementation and Evaluation Details

This appendix provides supplementary details regarding the implementation of baseline methods, SOTA models, and the evaluation procedure used in our experiments, complementing the core methodology described in Sections 3.2 and 4.3.

B.1 Lexical Baselines (Hashing, TF-IDF, Count) Implementation Details

Vectorizers: We used implementations from scikit-learn[11]. All vectorizers were configured with `lowercase=True`.

- `TfidfVectorizer` and `CountVectorizer`: Used an `ngram_range=(1, 2)`. Their vocabulary was constructed by first collecting unique tokens from all columns across the entire corpus (query tables included), ensuring a consistent feature space.
- `HashingVectorizer`: Used an `ngram_range=(1, 1)` and `alternate_sign=False`.

Input Data: For each table, we randomly sampled up to 1000 unique non-null cell values per column.

Vectorization: Each column's sampled values were treated as a document and vectorized into a 4096-dimensional vector using the appropriately fitted or configured vectorizer.

B.2 SOTA Method Implementation Details

B.2.a) Starmie: [12] We utilized the implementation and recommendations from the original Starmie paper (Fan et al., 2023b).

[11] https://scikit-learn.org/stable/api/sklearn.feature_extraction.html

[12] Starmie GitHub Repository

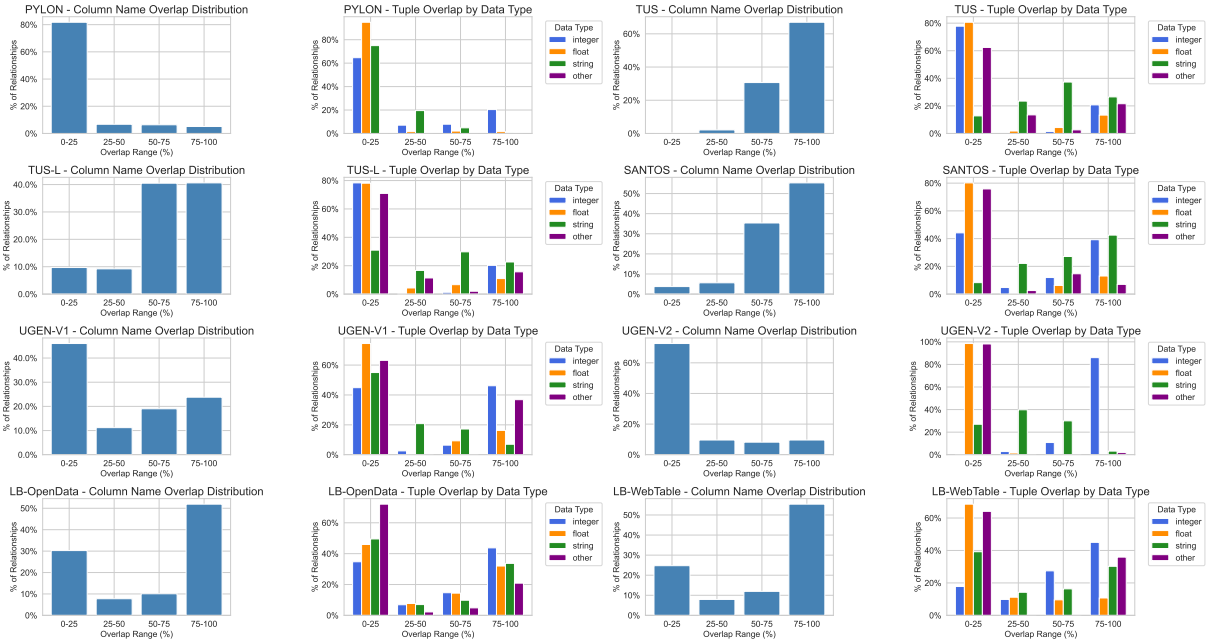

Figure 2: Distribution of exact column name and tuple overlap across different benchmarks, broken down by data type (String, Numeric, Datetime, Other). Each subplot represents a benchmark, showing the percentage of ground truth pairs falling into different overlap ranges.

Training Setup: The provided RoBERTa-based model was retrained for 10 epochs on each benchmark. Key hyperparameters included: batch size 32, projection dimension 768, learning rate 5e-5, max sequence length 256, and fp16 precision.

Sampling and Augmentation Strategies: Starmie employs specific strategies during contrastive pre-training to generate positive pairs (views of the same column). The strategies, based on the definitions in the original paper, are:

- *TF-IDF Entity Sampling (‘tfidf_entity‘):* Samples cells in columns that have the highest average TF-IDF scores calculated over their tokens.
- *Alpha Head Sampling (‘alphaHead‘):* Samples the first N tokens sorted alphabetically.
- *Column Dropping Augmentation (‘drop_col‘):* Creates augmented views by dropping a random subset of columns from the table.
- *Drop Cell Augmentation (‘drop_cell‘):* Creates augmented views by dropping random cells within the table.

We followed the paper’s recommendations for each benchmark, detailed in Table 6. For benchmarks not explicitly mentioned in the original paper (PYLON, UGEN, LAKEBENCH derivatives), we applied the same strategies recommended for the SANTOS benchmark.

Evaluation: We used the "Pruning" search strategy described in the Starmie paper, also referred to as "bounds" in the original implementation. This involves a maximum bipartite matching approach on a pruned set of candidate column pairs to calculate table similarity, offering higher efficiency compared to naive matching, while remaining more precise than approximate search approaches.

Benchmark	Sampling	Augmentation
SANTOS	tfidf_entity	drop_col
TUS (Small)	alphaHead	drop_cell
$\text{TUS}_{\text{Large}}$	tfidf_entity	drop_cell
PYLON	tfidf_entity	drop_col
UGEN_{V1}	tfidf_entity	drop_col
UGEN_{V2}	tfidf_entity	drop_col
LB-OPENDATA	tfidf_entity	drop_col
LB-WEBTABLE	tfidf_entity	drop_col

Table 6: Starmie sampling and augmentation strategies applied per benchmark.

B.2.b) HEARTS: [13]

Model: Employs pre-trained HyTrel embeddings (Chen et al., 2023), utilizing a publicly available checkpoint trained with a contrastive learning objective[14]. No further finetuning was performed.

[13] HEARTS GitHub Repository

[14] https://github.com/awslabs/hypergraph-tabular-lm/tree/main/checkpoints

Evaluation Strategy: We adopted the best-performing search strategy reported in the HEARTS repository for each benchmark:

- **Cluster Search (for SANTOS, PYLON, UGEN_{V1}, UGEN_{V2}):** This strategy first reduces the dimensionality of the pre-trained HyTrel column embeddings using UMAP (McInnes et al., 2018) and then performs clustering using HDBSCAN (McInnes et al., 2017). Default parameters provided in the HEARTS repository were used for both UMAP and HDBSCAN within this search method. Table similarity is derived based on cluster assignments.
- **FAISS + Max Pooling (for $\text{TUS}_{\text{Small}}$, $\text{TUS}_{\text{Large}}$, LB-OPENDATA, LB-WEBTABLE):** This strategy uses FAISS (Douze et al., 2024) for efficient similarity search. Table vectors are computed by max-pooling the embeddings of their constituent columns before indexing and searching.

B.3 Hardware

Our experiments were conducted using the following setup:

- CPU: Intel Xeon Gold 6330: 4 cores / 8 threads @ 2.00 GHz.
- GPU: 40GB MIG partition of NVIDIA A100 (used for SBERT embedding generation and SOTA models training/inference).
- 64 Go DDR4 RAM.

C Inconsistent Ground Truth Examples

This section provides illustrative examples of the ground truth inconsistencies identified in the UGEN and LAKEBENCH benchmarks during our analysis (Section 5.2). We categorize these into False Positives (pairs incorrectly labeled as unionable) and False Negatives (pairs incorrectly labeled as non-unionable or missed).

C.1 UGEN Benchmark Inconsistencies

Figures 3 and 4 showcase examples from UGEN variants.

C.2 Lakebench Benchmark Inconsistencies

This subsection presents examples of GTFPs from the LAKEBENCH benchmarks, where semantically and structurally compatible tables were not labeled as unionable in the ground truth but were correctly retrieved by search methods. Figures 5 and 6 show such cases from the WebTable and OpenData subsets, respectively.

Query: Anthropology_FGTNBDWF.csv
Candidate: Anthropology_N30U114M.csv

Age	Culture	Arena	Domain	Meaning	Origin	Activity
1	Neo.	Arch.	Past	Prim.	Africa	Hunt.
2	Islam.	Artif.	Hist.	Cplx.	Asia	Farm.

Artifact	Language	Technology	Education	Society
1	English	GPS	Political	Communal
2	Latin	Smartphone	Scientific	Global

(a) UGEN_{V1} Example: Tables discussing structurally and semantically distinct aspects of Anthropology (historical cultures vs. social technology), originally labeled unionable despite conceptual incompatibility.

Query: Anthropology_N7BS08I4.csv
Candidate: Anthropology_VS4SJ2VH.csv

Site Name	Location	Period	Culture
Olduvai Gorge	Tanzania, Africa	Pliocene	Hominin
Teotihuacan	Central Mexico	Early Classic	Teotihuacanos

Age Group	Clothing	Food	Housing
Children (0-12)	Tunics, hides	Porridge, roots	Huts (branch)
Teenagers (13-19)	Garments, beads	Grains, stews	Huts (woven)

(b) UGEN_{V2} Example: Tables about archaeological sites versus demographic lifestyles, representing fundamentally different entity types despite the shared Anthropology topic.

Figure 3: Examples of UGEN where pairs labeled unionable in the original ground truth exhibit significant semantic/structural divergence suggesting non-unionability.

D LLM Adjudicator

D.1 Prompt Details

To systematically re-evaluate potential ground truth inconsistencies in the UGEN benchmarks, we employed an LLM-based adjudicator. This process targeted disagreements identified during our analysis, specifically Ground Truth False Positives (GTFPs, pairs retrieved as potentially unionable within a rank threshold k' but not labeled as unionable in the ground truth, $k' < k$) and Ground Truth False Negatives (GTFNs, pairs labeled as unionable in the ground truth but retrieved within a rank threshold k', $k' > k$, or not retrieved at all).

For each query-candidate pair under review, we provided the LLM with the full content of both tables. The table data was serialized into a Markdown format using the `MarkdownRawTableSerializer` recipe from the Table Serialization Kitchen library[15](Gomm and Hulsebos, 2025). This serialized data was inserted into specific placeholders (`‘<Query Table Data>‘`, `‘<Candidate Table Data>‘`) within

[15]Table Serialization Kitchen Github Repository

Query: Archeology_2LWSQ5A2.csv
Candidate: Archeology_3ML53C0M.csv

Discovery	Item	Artifact	Date	Culture	Region
Giza Pyramid	Scroll	Diamond	~2500 BC	Anc. Egypt	N. Africa
Tut. Tomb	Knife	Stone Tab.	1323 BC	Anc. Egypt	N. Africa

Item	Discovery	Artifact	Date	Culture	Region
Scroll	Giza Pyramid	Diamond	~2500 BC	Anc. Egypt	N. Africa
Knife	Tut. Tomb	Stone Tab.	1323 BC	Anc. Egypt	N. Africa

(a) UGEN_{V1} Example: Two archaeology tables with identical information and permuted but perfectly alignable columns, incorrectly labeled non-unionable despite clear semantic compatibility.

Query: Veterinary-Science_YP1NJGLN.csv
Candidate: Veterinary-Medicine_GVNM098Q.csv

Animal Type	Breed	Age	Health Status	Symptoms	Diagnosis
Dog	Labrador Retr.	3 years	Healthy	No symptoms	Routine check-up
Cat	Domestic SH	5 years	Overweight	Lethargy...	Obesity

Animal Type	Breed	Age	Gender	Symptoms	Diagnosis
Dog	Labrador	3 years	Male	Aggression...	Rabies
Cat	Siamese	8 years	Female	Limping...	Arthritis

(b) UGEN_{V2} Example: Two veterinary case tables with highly alignable core columns (Animal Type, Breed, Age, Symptoms, Diagnosis) representing the same fundamental entity type (animal patients).

Figure 4: Examples of UGEN Pairs explicitly labeled as non-unionable in the original ground truth exhibiting strong compatibility suggesting unionability.

Query: csvData10212811.csv
Candidate: csvData1066748.csv

Player	Team	POS	G	AB	H	HR	...	OPS
B Dean	GL	1B	96	350	83	7	...	0.657
Y Arbelo	SB	1B	134	461	114	31	...	0.877

Player	Team	POS	G	AB	H	HR	...	OPS
J Colina	WS	2B	59	216	66	3	...	0.832
B Friday	LYN	SS	85	341	98	2	...	0.752

(a) WebTable Example 1: Baseball player statistics tables with identical, rich schemas (including Player, Team, POS, G, AB, H, HR, OPS, etc.). These tables represent the same entity type (player season stats) and are highly unionable, but were not labeled as such in the ground truth.

Query: csvData10025189.csv
Candidate: csvData20099586.csv

Player	Team	POS	AVG	G	AB	R	...	OPS
A Ramirez	MIL	3B	0.285	133	494	47	...	0.757
E Chavez	ARI	3B	0.246	44	69	6	...	0.795

Player	Team	POS	AVG	G	AB	R	...	OPS
L Castillo	NYM	2B	0.245	87	298	46	...	0.660
R Durham	MIL	2B	0.289	128	370	64	...	0.813

(b) WebTable Example 2: More baseball player statistics tables with identical schemas, clearly unionable but not labeled as such.

Figure 5: Examples of LB-WEBTABLE Ground Truth Incompleteness.

Source: OpenData (Canada)
Query: CAN_CSV0000000000000659.csv
Candidate: CAN_CSV0000000000000562.csv

REF_DATE	GEO	Age group	Sex	...	VALUE
2003	Canada	Total, 12 years and over	Both sexes	...	20723896.0
2003	Canada	Total, 12 years and over	Both sexes	...	20632799.0

REF_DATE	GEO	Age group	Sex	...	VALUE
2003	Canada	Total, 12 years and over	Both sexes	...	26567928.0
2003	Canada	Total, 12 years and over	Both sexes	...	26567928.0

(a) OpenData Example 1: Canadian health survey tables sharing key demographic columns (REF_DATE, GEO, Age group, Sex) for the same population. This pair represents unionable statistics about that population but was not labeled as unionable in the ground truth.

Source: OpenData (Canada)
Query: CAN_CSV0000000000000686.csv
Candidate: CAN_CSV0000000000005304.csv

Sex	Type of work	Hourly wages	UOM	UOM_ID	...	VALUE
Both	Both full- and part...	Total employees, all wages	Persons	249	...	10921.0
Males	Both full- and part...	Total employees, all wages	Persons	249	...	5645.4

Sex	Type of work	Weekly wages	UOM	UOM_ID	...	VALUE
Both	Both full- and part...	Total employees, all wages	Persons	249	...	11364.5
Males	Both full- and part...	Total employees, all wages	Persons	249	...	5954.5

(b) OpenData Example 2: Canadian employment statistics. The query table (data related to 'Hourly wages') and candidate table (data related to 'Weekly wages') share key dimensions like Sex, Type of work, and UOM. The cell values within their respective 'Hourly wages'/'Weekly wages' columns (e.g., 'Total employees, all wages') describe similar employee groups. This pair, differing mainly in wage aggregation period (hourly vs. weekly) and slightly in REF_DATE format (YYYY vs YYYY-MM), is potentially unionable for comprehensive wage analysis but was not labeled as such in the ground truth.

Figure 6: Examples of LB-OPENDATA Ground Truth Incompleteness.

the prompt detailed below. Crucially, the original table names were *not* included in the prompt. This decision was made to avoid potentially biasing the LLM by providing explicit hints about the table's topic beforehand, thereby ensuring the adjudication relies solely on the semantic and structural information present in the table content itself.

The prompt utilizes few-shot learning, incorporating hand-selected positive and negative examples of unionability from the UGEN benchmarks themselves to guide the LLM's judgment (these examples are represented by a placeholder in the verbatim prompt below for brevity). The prompt defines the LLM's role, outlines core principles for assessing conceptual coherence and semantic column alignment, and specifies the required output format.

The complete prompt structure provided to the LLM adjudicator is shown below:

```
You are an experienced data curator evaluating
    if two database tables can be meaningfully
    combined vertically (unioned). The goal of
    unioning is to create a single, larger
    dataset containing the same kind of
    information or describing the same type of
    entity/event.

Your task is to determine if TABLE 1 and TABLE
    2 are conceptually compatible enough for a
    union operation.

CORE PRINCIPLES FOR UNIONABILITY:

1. Conceptual Coherence: Do both tables
    fundamentally describe the same type of
    entity (e.g., customers, products, logs) or
    record the same type of event (e.g., sales,
    website visits)? Appending rows from one
    table to the other should result in a
    dataset that makes logical sense.

2. Meaningful Column Alignment: There must be a
    reasonable set of columns across the two
    tables that represent the same underlying
    attributes or concepts.
   * These columns can have DIFFERENT NAMES
       (e.g., "Cust_ID" vs.
       "ClientIdentifier").
   * They can have DIFFERENT FORMATS (e.g.,
       "2023-01-15" vs. "1/15/2023").
   * They may have LITTLE TO NO OVERLAP in
       actual data values.
   * Focus on the semantic meaning of the
       columns in the context of their
       respective tables.

3. Sufficient Column Matching: The alignment
    shouldn't rely on just one incidental or
    minor column. There should be enough
    matching among key columns to confidently
    conclude that the tables represent the same
    underlying information. More aligned
    columns representing core attributes
    increase confidence.

4. Distinction from Joins: We are NOT looking
    for keys to join tables horizontally. We
    are assessing if they can be stacked
    vertically.

[EXAMPLES OF UNIONABILITY/NON-UNIONABILTY:]
(Hand-selected positive/negative examples from
    \textsc{Ugen} benchmark were inserted here
    in the actual prompt)

YOUR TASK:

Examine the following two tables based on the
    principles and examples above:

TABLE 1:
<Query Table Data (Serialized)>

TABLE 2:
<Candidate Table Data (Serialized)>

PROVIDE YOUR ANSWER IN THE FOLLOWING FORMAT:

1. First line: `UNIONABLE: Yes` or `UNIONABLE:
    No`
2. Second line: `EXPLANATION:` followed by a
    brief justification focusing on the
    conceptual coherence and the sufficiency of
    semantic column alignment. Explain why they
    are or are not the same kind of data.
```

D.2 Example LLM Outputs

Below are illustrative examples of the outputs generated by the LLM adjudicator based on the prompt above for some table pairs showcased in Appendix C.1.

Example 1: Archeology_2LWSQ5A2.csv vs. Archeology_3ML53C0M.csv (UGEN-V1)

```
UNIONABLE: Yes
EXPLANATION: Both tables describe archeological
    artifacts and their attributes (discovery
    location, item type, date, culture,
    material, etc.). They represent the same
    kind of data - records of archeological
    findings. Unioning them would create a
    consolidated dataset of such findings.
    There is strong semantic alignment across
    almost all columns, even though the column
    order differs.
```

Example 2: Anthropology_FGTNBDWF.csv vs. Anthropology_N30U114M.csv (UGEN-V1)

```
UNIONABLE: No
EXPLANATION: TABLE 1 categorizes
    anthropological concepts based on
    dimensions like 'Age', 'Culture', 'Arena',
    'Domain', 'Meaning', 'Origin', and
    'Activity'. TABLE 2, on the other hand,
    lists categories such as 'Artifact',
    'Language', 'Technology', 'Education', and
    'Society' as columns. While both tables are
    related to Anthropology, they represent
    different perspectives and categorizations
    within the field. There is insufficient
    semantic overlap in the columns to suggest
    they describe the same type of entity or
    event. Unioning them would likely result in
    a table that is not conceptually coherent,
    as the rows from each table represent
    different kinds of anthropological
    classifications.
```