

LLM-Driven Retrieval, Debate, and Verification for Robust Table-to-Knowledge-Graph Matching

Koby Bar^{1,*}, Tomer Sagi²

¹University of Haifa, Haifa, Israel

²Aalborg University, Aalborg, Denmark

Abstract

Tabular data is one of the most common data sources on the internet and is widely used in various data analytics tasks. Identifying semantic concepts within tables is often a critical component of these pipelines, yet it remains a challenging task to automate. To address this problem, we present RAGDify, a large language model (LLM)-based system designed for the Cell Entity Annotation (CEA) task. Our system employs a three-step pipeline inspired by Retrieval-Augmented Generation (RAG) and advanced reasoning techniques: (1) retrieving context-aware candidate entities, (2) engaging in a debate-like evaluation to compare top candidates, and (3) applying chain-of-verification-inspired prompting to validate the final entity match. We propose RAGDify as a solution for the SemTab'25 challenge, targeting the key challenges inherent in automating the CEA task.

Keywords

Cell Entity Annotation, Table to Knowledge Graph Matching, Large Language Models Reasoning, Entity Matching, Retrieval Augmented Generation

1. Introduction

In recent years, we have witnessed a significant increase in the availability and dissemination of data, particularly tabular data [1]. Many processes have become data-driven, requiring large volumes of structured data to train algorithms or support decision-making. Tabular formats such as CSV have become the de facto standard for representing and exchanging data due to their compact, human-readable structure.

Automatically processing tabular data is a fundamental step in a wide range of applications, including schema matching, entity linking, question answering, and knowledge graph construction. A common approach to facilitate this processing is to link table elements, such as cells, or columns, to entities and concepts within a Knowledge Graph (KG) or ontology. This linkage creates a semantic layer that enables higher-level reasoning and integration across heterogeneous datasets—a process commonly referred to as *Semantic Table Interpretation (STI)* [2].

Despite its widespread use and simplicity, the tabular format poses several challenges. Tables often lack explicit contextual information, exhibit semantic ambiguities, and are susceptible to inconsistencies and noise in their data. Consequently, automating the semantic interpretation of tables remains an open and complex research problem [3].

One promising direction to address these challenges is the incorporation of Large Language Models (LLMs) into the Semantic Table Annotation pipeline [4]. LLMs, trained on vast amounts of textual data, have demonstrated impressive abilities in solving tasks beyond their specific training objectives, even in settings with limited annotated data (few-shot learning [5]) or no task-specific data at all (zero-shot learning [6]).

However, directly applying LLMs to STI tasks introduces several key obstacles. First, LLMs are not explicitly trained on structured KG data, and thus struggle with complex entity disambiguation tasks

SemTab'25: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching 2025, co-located with the 24th International Semantic Web Conference (ISWC), November 02–06, 2025, Nara, Japan

*Corresponding author.

✉ barkoby@gmail.com (K. Bar); tsagi@cs.aau.dk (T. Sagi)

ORCID 0000-0002-6354-422X (K. Bar); 0000-0002-8916-0128 (T. Sagi)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

where numerous KG entities share identical or highly similar surface forms. Second, the inherent lack of context within tabular structures exacerbates the difficulty of semantic interpretation. Third, LLMs are prone to hallucinations, producing confident but factually incorrect outputs, which can severely undermine the accuracy of entity annotations [7].

To mitigate these challenges, a *Retrieval-Augmented Generation (RAG)* architecture has emerged as a promising solution [8]. RAG architectures enhance LLMs by grounding their responses in external knowledge, limiting the candidate space to a set of relevant entities retrieved from a KG, and guiding the LLM through the entity matching process. This retrieval-augmented approach not only constrains the model’s generation space but also improves its factual reliability.

Moreover, integrating advanced reasoning techniques has shown potential in boosting LLM factual accuracy and decision-making capabilities. Recent advancements include multi-agent debate frameworks [9], chain-of-thought prompting [10], self-consistency mechanisms [11], and chain-of-verification [12] strategies that iteratively validate generated outputs against factual sources. These reasoning techniques have been effective in guiding LLMs through complex multi-step tasks, fostering both robustness and interpretability.

In this paper, we present our approach to the Cell Entity Annotation (CEA) task for the SemTab’25 Challenge—*RAGDify*. Our system leverages LLMs within a RAG-based architecture, enriched with a reasoning mechanism inspired by multi-agent debate, self-consistency, and chain-of-verification techniques.

The remainder of this paper is organized as follows: Section 2 provides an overview of the task and foundational approaches; Section 3 details our proposed methodology; Section 5 reviews related work in the field; finally, Section 6 highlights key challenges, summarizes our contributions and limitations, and outlines directions for future research.

2. The Task

2.1. Overview of the Challenge

The SemTab challenge started in 2019 with the goal to promote research in STI and provide a venue for benchmarking different solutions [4]. Over the years a wide range of solutions have been suggested for SemTab tasks. With the emergence of large language models (LLMs), special interest arose, leading to a dedicated *STI vs. LLMs Track* in 2024, and in 2025 all participants are expected to use LLM-based methods, either via fine-tuning or RAG.

We focus on the CEA task, which involves linking each table cell to its corresponding entity in a knowledge base (e.g., a KG or ontology). Figure 1 illustrates this process. Systems are evaluated using standard precision, recall, and F1-score metrics. In addition, the challenge requires that solutions address several key challenges, such as disambiguation, homonymy, alias resolution, NIL detection, noise robustness, and collective inference, to reflect the complexities of real-world table data.

2.2. MammoTab Track and Dataset

We competed in the MammoTab track, which leverages the most recent version of the MammoTab dataset [13]. MammoTab comprises 870 heterogeneous tables—sourced from Wikipedia, collectively contains 84,907 verified cell–entity annotations. This benchmark simulates the challenges of real-world table interpretation: tables lack explicit schema definitions, and cell contexts exhibit varying degrees of noise, ambiguity, and sparsity. For the MammoTab track, Wikidata (v. 20240720) serves as the target KG.

3. Methods

The *RAGDify* system formulates table-to-knowledge-base matching as a four-stage retrieval-generation pipeline (Figure 2), wherein each stage leverages a large language model (LLM) to balance high recall with precise disambiguation.

col0	col1	col2	col3	col4	col5
2012	California 33	Independent	Bill Bloomfield	146,660	2nd
2012	Colorado 5	Independent	Dave Anderson	53,318	2nd

California State Route 33 (Q662907), highway in California	CERRO COLORADO 5 (Q105970601), archaeological site in Cordoba Province, Argentina
California's 33rd congressional district (Q225706), U.S. House district in western suburbs of Los Angeles, CA	Grand Canyon (Q118841), steep-sided canyon carved by the Colorado River in Arizona, United States
2024 California Proposition 33 (Q130614755), 2024 California referendum	Colorado's 5th congressional district (Q5148668), U.S. House district centered on Colorado Springs, CO

Figure 1: Illustration of the CEA task. Candidate entities from the Wikidata knowledge graph are retrieved for the cell value “California 33” (**bottom left**) and “Colorado 5” (**bottom right**). The correct entities are highlighted in blue.

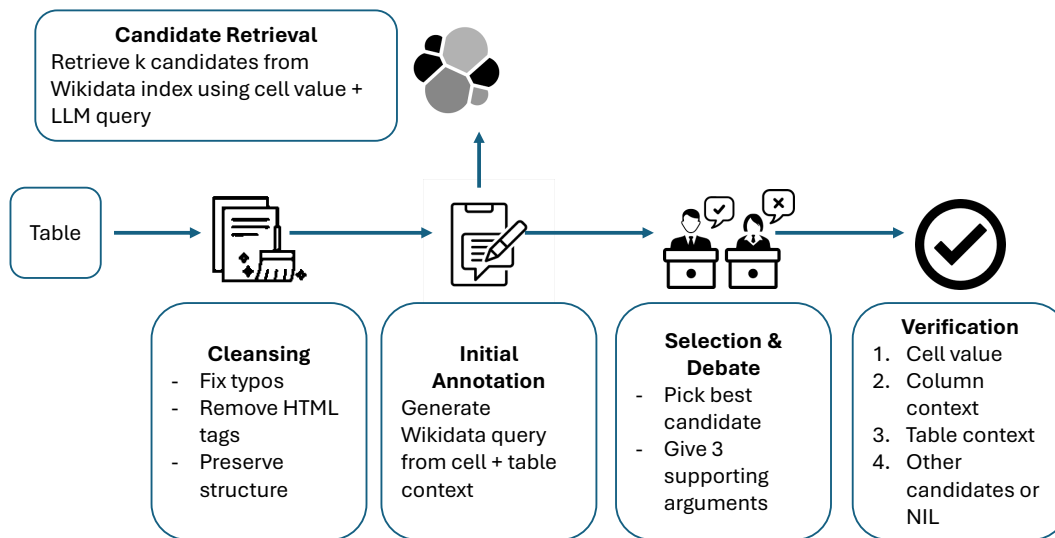


Figure 2: Overview of the RAGDify pipeline for cell entity annotation on Wikidata: (1) LLM-driven table cleansing; (2) candidate retrieval via Elasticsearch using exact match, LLM-rewritten contextual queries, and fuzzy lookup; (3) debate-style LLM ranking with three supporting arguments; (4) final LLM-based validation across cell, column, and table contexts, including NIL detection.

In the *data cleansing* stage, raw CSV tables are ingested and lightly cleaned via an LLM prompt that corrects typographical or formatting errors, unifies casing, and removes noise, e.g., stray punctuation or outlier tokens, while preserving the original row and column structure. This preprocessing yields a consistent set of cell values for downstream retrieval.

The *candidate generation* stage employs three sequential retrieval strategies over a full-text inverted index of Wikidata entity labels and descriptions, implemented in Elasticsearch. First, we perform an index lookup using the cleaned cell string. Second, regardless of whether the first lookup succeeded, we invoke a few-shot LLM prompt to reformulate the query by incorporating column and table meta-data—capturing aliases, synonyms, and contextual nuances. These results are added to the candidate list. Finally, if neither of the first two strategies returns any results, we execute a fuzzy search requiring at least 75% string similarity. To bound LLM context and reduce latency, tables with more than ten rows

Candidate Generation

Direct lookup:

California State Route 33 (Q662907), California's 33rd congressional district (Q225706), Riverside County (Q108111), and 2024 California Proposition 33 (Q130614755)

LLM-reformulated query: California 33 district:

California's 33rd State Assembly district (Q5020024), California's 33rd congressional district (Q225706), California's 33rd State Senate district (Q5020025), and San Joaquin County Sheriff's Department (Q7414388).

Debate and Select

URI: <http://www.wikidata.org/entity/Q225706>

Arguments:

1. The candidate label "California's 33rd congressional district" exactly matches the cell value "California 33," indicating the same federal electoral district.
2. Column 2 ("Independent") and Column 3 ("Bill Bloomfield") refer to the party and representative of a congressional district, consistent with this URI.
3. The population (146 660) and rank ("2nd") in Columns 4–5 align with demographic metrics typically reported for congressional districts, reinforcing the match.

Verification

Verification: yes

Winning candidate URI: <http://www.wikidata.org/entity/Q225706>

Figure 3: Illustrative example of RAGDify's CEA pipeline for the cell value *California 33*. **Left:** Candidate Generation combining a direct lookup and few-shot LLM reformulation to assemble the candidate set. **Right:** Debate & Select prompt with URI selection and supporting arguments. **Bottom:** The LLM answer to the verification prompt to confirm the final match against the verification questions.

are truncated to the ten rows nearest the target cell.

In the *candidate ranking* stage, we employ a debate-style prompting strategy in which the LLM receives a set of candidates and is asked to nominate the most plausible entites, accompanied by three concise, evidence-based arguments referencing the cell value and its surrounding context. This argumentative framing, rather than relying solely on similarity scores, encourages the model to surface the strongest semantic match. To balance cost and accuracy, the debate can be run over the entire candidate set or restricted to a top- k subset, and it may be iterated for multiple rounds, with each iteration refining the argumentation and narrowing the pool until a final winner emerges.

Finally, the *validation* stage tasks the LLM with targeted questions probing the chosen entity's consistency with the original cell value, compatibility with column and table context, and distinction from alternative candidates, including an explicit NIL option. Based on these responses, the system either confirms the selection or revises it and may iterate the validation prompts for several rounds until a stable, final annotation is produced. Figure 4 shows the key LLM prompt templates used in the pipeline's candidate generation, candidate ranking, and validation stages. Figure 3 illustrates the main stages applied to the CSV example from Figure 1.

This pipeline tackles all core CEA challenges in one sweep: by normalizing and denoising cells up front, it achieves noise robustness; its three-stage retrieval (exact, contextual reformulation, fuzzy) and debate-style ranking enforce both disambiguation and homonymy resolution using column and table level cues; contextual query rewriting surfaces aliases and nicknames; finally, because both reformulation and validation draw on neighboring cells, our approach produces coherent, table-wide annotations.

Implementation Details The proposed pipeline is LLM-agnostic and can be adapted with minimal modifications to a variety of models. Given the relatively large size of the test set (see Section 2.2), we prioritized runtime and cost efficiency. To this end, during ranking we generate supporting arguments only for the top-ranked candidate, avoiding per-candidate debates, and perform a single verification step. For our experiments, we chose OpenAI's *GPT-4.1 nano* due to its favorable cost-to-performance ratio.

All experiments were conducted on an Ubuntu 20.04.6 Linux server equipped with two Intel Xeon Gold 6326 CPUs (16 cores per socket, 2 sockets, 64 threads total, 2.90 GHz) and 256 GB of RAM. The entire pipeline, including the LLM client, retrieval modules, and validation logic, was containerized using Docker and orchestrated via Docker Compose. Elasticsearch was deployed in a dedicated Docker container, and GPT API calls were parallelized with a 4-thread pool to maximize throughput while

Candidate Generation	Debate and Select	Verification
Given a CSV table and a target cell ({row_id}, {col_id}, {value})	Given a target cell in a CSV table ({row_id}, {col_id}, {value}) and a list of candidate entities ({candidates}) from Wikidata	Re-evaluate the selected candidate using the table, candidate list, and arguments
<ul style="list-style-type: none"> - Generate a search query for Wikidata. Few-shot examples: {examples} - Consider abbreviations, synonyms, and variations. 	<ul style="list-style-type: none"> - Select the best match for the cell, provide at least 3 strong arguments supporting your choice - Consider the table context 	<ul style="list-style-type: none"> - Check fit with cell value, column values, and table context - Revise if a better candidate exists; otherwise confirm the choice. Use NIL if no candidate fits
Output only the search text	Output format: URI: <candidate URI> Arguments: <arguments>	Output format: Verification: <yes/no> Winning candidate URI: <candidate URI or NIL>

Figure 4: Key LLM prompt templates used in RAGDify’s CEA pipeline. **Left:** Candidate generation prompt, which presents the CSV context and instructs the model to suggest a concise search string for the Wikidata index. **Middle:** Debate and select prompt, which lists candidate URIs with labels and descriptions and asks the LLM to choose one (or more) candidates that best match the cell, with three supporting arguments. **Right:** Verification prompt, which provides the selected candidate, its arguments, and the full candidate list, then asks the LLM to reconsider its choice by probing cell-, column-, and table-level consistency and to confirm or revise the final URI (or choose NIL).

respecting rate limits. End-to-end processing of the SemTab’25 test set required approximately 26 hours and incurred US\$26.60 in API costs.

4. Results

Table 1 shows the SemTab’25 results on the MammoTab track.

Table 1

Performance comparison on the MammoTab benchmark from the official leaderboard [14].

Model	Precision	Recall	F1 Score
RAGDify	0.603	0.603	0.603
ADFr	0.758	0.758	0.758
ditlab	0.549	0.549	0.549
Kepler-aSI	0.403	0.157	0.226

5. Related Work

Tabular data linking to a KG has been studied for over two decades. Early STI pipelines typically comprised three sequential stages: pre-processing (cleaning and denoising), candidate generation via keyword- or schema-based lookup, and iterative disambiguation to resolve noise and ambiguity [3].

Initial LLM-based approaches to CEA focused on learning joint embeddings of table cells and KG entities. TURL [15] extends TinyBERT with a structure-aware visibility matrix, jointly optimizing masked language modeling and masked entity retrieval objectives. TableLlama [16], a decoder-only variant of Llama 2-7B, employs LongLora to accommodate extended contexts (up to 8,192 tokens) and is instruction-tuned on a large TableInstruct corpus for CEA and other table tasks. TAPAS [17] adapts BERT with table-specific position embeddings and aggregation heads to perform both cell classification and entity linking, achieving strong in-domain performance without external KG queries.

Replacing or unifying traditional STI stages, prompting over large generative models has become popular. TSOTSA [18] leverages GPT-based prompts for candidate retrieval and ranking; Kepler-aSI [19] integrates SPARQL query outputs into LLM inference to refine entity selection; CitySTI [20] applies end-to-end prompting across all STI phases; and Adwan [21] demonstrates a retrieval-augmented

generation (RAG) pipeline enhanced with chain-of-thought and self-consistency prompting for robust table metadata linking.

While joint-representation models like TURL, TAPAS, and TableLlama excel at in-domain embedding efficiency, they often require substantial labeled data or fine-tuning. Prompting-based methods simplify deployment and achieve strong zero-shot performance but can incur higher API costs and latency. Our RAGDify system builds on these paradigms by combining LLM-driven query reformulation, debate-style ranking, and explicit validation to deliver a versatile, cost-effective solution across all CEA challenges.

6. Conclusions

This paper has presented *RAGDify*, a retrieval-augmented generation pipeline for cell entity annotation. Our approach combines four key components: (1) lightweight LLM-driven table cleansing to correct typos and normalize values; (2) multi-stage candidate retrieval via exact match, contextual (LLM-rewritten) queries, and fuzzy lookup to maximize recall; (3) debate-style ranking that prompts the LLM to select a single top candidate with supporting arguments; and (4) explicit validation that probes cell-, column-, and table-level consistency and allows for NIL assignments. By design, RAGDify is LLM-agnostic, adapts with minimal changes to different model families, and maintains cost efficiency through a single debate round and one verification step.

Future Work Several directions merit further exploration. First, a controlled study of debate and verification depths, varying the number of argumentation rounds and follow-up checks—could identify the optimal balance between annotation accuracy and computational cost. Second, integrating a learned semantic retrieval layer (e.g., dense embeddings) promises to boost candidate recall beyond syntactic lookup without a significant runtime penalty. Third, access to a high-quality gold annotation set for the SemTab’25 benchmark or another comparable dataset would enable rigorous evaluation and targeted fine-tuning. Such a dataset would also allow a detailed analysis of how each component (retrieval, debate, verification) contributes to overall performance, potentially closing the gap with fully supervised methods.

Limitations Despite its advantages, the proposed pipeline has several limitations. First, it relies on syntactic search over an Elasticsearch index, which may limit recall; integrating a robust semantic search could substantially improve retrieval performance. Second, the method is entirely prompt-based, primarily zero-shot, to maintain dataset agnosticism; this design choice can be suboptimal compared to task-specific fine-tuning, which typically yields higher annotation accuracy. Third, to control cost, the debate verification mechanism is limited to a single round; extending it to multiple stages could further enhance matching quality.

7. Acknowledgments

This work was supported by the Data Science Research Center at the University of Haifa through the Israel PBC grant Advancing Data Science to Serve Humanity and Protect the Global Environment (grant no. 100009443).

Declaration on Generative AI

During the preparation of this manuscript, the authors used OpenAI’s o4-mini to correct grammatical errors and spelling mistakes. As described in Section 3, GPT-4.1 nano served as the primary LLM for our method. Microsoft Copilot (powered by GPT-4o) was employed as a coding assistant during model development. All outputs generated by these tools were critically reviewed and edited by the authors, who take full responsibility for the content of this publication.

References

- [1] O. Benjelloun, S. Chen, N. Noy, Google dataset search by the numbers, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12507 LNCS (2020) 667–682. URL: https://link.springer.com/chapter/10.1007/978-3-030-62466-8_41. doi:10.1007/978-3-030-62466-8_41.
- [2] M. Cremaschi, B. Spahiu, M. Palmonari, E. Jimenez-Ruiz, Survey on semantic interpretation of tabular data: Challenges and directions, *arXiv preprint* (2024). URL: <https://arxiv.org/pdf/2411.11891>.
- [3] J. Liu, Y. Chabot, R. Troncy, V. P. Huynh, T. Labbé, P. Monnin, From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods, *Journal of Web Semantics* 76 (2023) 100761. URL: <https://dl.acm.org/doi/10.1016/j.websem.2022.100761>. doi:10.1016/J.WEBSEM.2022.100761.
- [4] O. Hassanzadeh, N. Abdelmageed, M. Cremaschi, V. Cutrona, F. D’adda, V. Efthymiou, B. Kruit, E. Lobo, N. Mihindukulasooriya, N. H. Pham, Results of semtab 2024, 2024. URL: <https://ceur-ws.org/Vol-3889/paper0.pdf>.
- [5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. Mccandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, *Advances in Neural Information Processing Systems* 33 (2020) 1877–1901. URL: <https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf>.
- [6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, *OpenAI Blog* (2019).
- [7] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, P. Fung, Survey of hallucination in natural language generation, *ACM Computing Surveys* 55 (2023). URL: <https://dl.acm.org/doi/pdf/10.1145/3571730>. doi:10.1145/3571730/ASSET/CC5D3792-8BC0-4675-8584-B507476E20EC.
- [8] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, *Advances in Neural Information Processing Systems* 33 (2020) 9459–9474. URL: <https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf>.
- [9] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, I. Mordatch, Improving factuality and reasoning in language models through multiagent debate, *CoRR abs/2305.14325* (2023).
- [10] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. C. Quoc, V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, *Advances in Neural Information Processing Systems* 35 (2022) 24824–24837.
- [11] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. H. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, *11th International Conference on Learning Representations, ICLR 2023* (2023).
- [12] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, J. Weston, Chain-of-verification reduces hallucination in large language models, *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (2024) 3563–3578. URL: <https://aclanthology.org/2024.findings-acl.212/>. doi:10.18653/V1/2024.FINDINGS-ACL.212.
- [13] M. Marzocchi, M. Cremaschi, R. Pozzi, R. Avogadro, M. Palmonari, Mammotab: a giant and comprehensive dataset for semantic table interpretation, in: *Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching, SemTab 2022, co-located with the 21st International Semantic Web Conference (ISWC), CEUR-WS.org, 2022*. URL: <http://ceur-ws.org>.
- [14] SemTab Challenge Organizers, Semtab 2025 leaderboard, <https://sem-tab-challenge.github.io/2025/#leaderboard>, 2025. Accessed: October 21, 2025.
- [15] X. Deng, H. Sun, A. Lees, Y. Wu, C. Yu, Turl: Table understanding through representation learning, *SIGMOD Record* 51 (2022) 33–40. doi:10.1145/3542700.3542709.

- [16] T. Zhang, X. Yue, Y. Li, H. Sun, Tablellama: Towards open large generalist models for tables, in: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2024, volume 1, Association for Computational Linguistics (ACL), 2023, pp. 6024–6044. doi:10.18653/v1/2024.naacl-long.335.
- [17] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, J. M. Eisenschlos, Tapas: Weakly supervised table parsing via pre-training, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics (ACL), 2020, pp. 4320–4333. URL: <https://aclanthology.org/2020.acl-main.398/>. doi:10.18653/v1/2020.ACL-MAIN.398.
- [18] J. P. Bikim, C. Atezong, A. Jiomekong, A. Oelen, G. Rabby, J. D. . Souza, S. Auer, Leveraging gpt models for semantic table annotation, in: SemTab’24: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching 2024, co-located with the 23rd International Semantic Web Conference (ISWC), 2024.
- [19] W. Baazouzi, M. Kachroudi, S. Faiz, Kepler-asi : Semantic annotation for tabular data, in: SemTab’24: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching 2024, co-located with the 23rd International Semantic Web Conference (ISWC), 2024.
- [20] D. Li, T. Yue, E. Jimenez-Ruiz, Citysti 2024 system: Tabular data to kg matching using llms, in: SemTab’24: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching 2024, co-located with the 23rd International Semantic Web Conference (ISWC), 2024.
- [21] N. Vandemoortele, B. Steenwinckel, S. V. Hoecke, F. Ongenae, Scalable table-to-knowledge graph matching from metadata using llms, in: SemTab’24: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching 2024, co-located with the 23rd International Semantic Web Conference (ISWC), 2024.