Wikidata-Driven CEA and CTA for Life Sciences Table Matching extending DREIFLUSS

Vishvapalsinhji Parmar[1], Alsayed Algergawy[1]

[1]Chair of Data and Knowledge Engineering, University of Passau Passau, Germany

Abstract

In our previous work for the SemTab 2023 challenge, we presented DREIFLUSS, a minimalist approach utilizing machine learning models and sampling techniques to tackle Column Property Annotation (CPA) and Column Type Annotation (CTA) tasks. Building on this groundwork, this paper shifts focus for the SemTab 2024 challenge by harnessing the semantic capabilities of the Wikidata knowledge graph to address Cell Entity Annotation (CEA) and CTA tasks. Our approach leverages optimized preprocessing and querying techniques with the Wikidata API [1], leading to significant improvements in the accuracy and efficiency of table annotations. We achieved F1 scores of 93.20% for CEA and 61.50% for CTA on the tBiodivL-Horizontal dataset, along with an F1 score of 92.50% for CEA on the tBiomedL-Horizontal dataset. These results highlight the promise of knowledge graph-based methods in refining table-matching processes, laying the groundwork for future research that combines machine learning techniques with knowledge graph-driven strategies to achieve more robust annotation outcomes.

Keywords

Table Matching, Cell Entity Annotation (CEA), Column Type Annotation (CTA), Knowledge Discovery, Data Integration

1. Introduction

Matching tables to knowledge graphs, a vital aspect of data integration and knowledge discovery, has gained significant attention due to the proliferation of digital information. It involves harmonizing information across different tables, which is crucial for extracting valuable insights. With millions of high-quality tables available on the Internet—a number that continues to rise due to advancements in automated data extraction and the growing reliance on structured data across various sectors, including business, academia, and government [1]—effective table matching is more important than ever.

The SemTab Challenge[1] has emerged as a leading competition that pushes the frontiers of table understanding and annotation. In the 2023 edition, we introduced DREIFLUSS, a minimalist approach that utilized machine learning models and strategic sampling techniques to address the tasks of Column Property Annotation (CPA) and CTA [2]. This approach demonstrated the effectiveness of using data-driven techniques to achieve high accuracy in semantic table

[1]https://www.wikidata.org/w/api.php

SemTab'24: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching 2024, co-located with the 23rd International Semantic Web Conference (ISWC), November 11-15, 2024, Baltimore, USA

vishvapalsinhji.parmar@uni-passau.de (V. Parmar); alsayed.algergawy@uni-passau.de (A. Algergawy)

0000-0002-4370-2729 (V. Parmar); 0000-0002-8550-4720 (A. Algergawy)

[1]https://www.cs.ox.ac.uk/isg/challenges/sem-tab/

annotations. Building on this foundation, the 2024 SemTab challenge presented an opportunity to explore a different dimension of table annotation by leveraging the semantic richness of knowledge graphs. In this work, we extend the DREIFLUSS methodology by utilizing the Wikidata knowledge graph to tackle the tasks of CEA and CTA. Unlike the previous machine learning-based approach [2], this paper focuses on using the Wikidata API to extract and integrate semantic labels, which significantly enhances the precision and efficiency of table annotations.

By employing a knowledge graph-driven strategy, our approach showcases the potential of semantic resources like Wikidata in refining table matching processes. This shift allows for the exploration of new methods in table annotation, underscoring the importance of adaptability and scalability in today's data-driven landscape [3, 4]. The insights gained from this exploration also lay the groundwork for future research that combines knowledge graph-based techniques with machine learning approaches to further improve table annotation outcomes. The rapid growth of structured data on the web presents both immense opportunities for knowledge discovery and significant challenges. Each table often comes with a unique structure, schema, and notation, requiring advanced methods for understanding and harmonization. Competitions like SemTab play a vital role in addressing these challenges by advancing the capabilities of table understanding and annotation. The critical tasks of CTA and CEA are central to achieving comprehensive table comprehension, efficient data integration, and effective knowledge discovery.

To address these needs, our current methodology leverages pre-existing semantic resources, specifically focusing on Wikidata to enhance the table annotation process. This approach demonstrates the advantages of using a knowledge graph-based strategy to improve annotation accuracy and efficiency. Moreover, it provides inspiration for future work that could integrate machine learning models with semantic resources to develop more robust and adaptable solutions for table annotation challenges. This work focuses on datasets from Life Sciences, such as those in biodiversity and biomedicine, where accurate table annotation is critical for knowledge discovery and data integration in domains like healthcare and biology.

2. Related Work

Since its inception in 2019, the SemTab challenge has been instrumental in advancing the field of semantic table interpretation, which focuses on understanding and annotating tabular data with semantic information. In the inaugural year, Oliveira and d'Aquin introduced "ADOG" [5], a system that utilizes ontologies for data annotation. Complementing this, Cremaschi et al. presented "MantisTable" [6], an innovative system for automatic semantic table interpretation. Another significant contribution was made by Thawani et al., who focused on CTA and CPA tasks, developing a method to link entities to knowledge graphs for inferring column types and properties [7].

The challenge evolved in 2020 with Huynh et al.'s enhanced version of "DAGOBAH" [8], which highlighted scalable annotations for large datasets. Concurrently, Abdelmageed and Schindler introduced "JenTab" [9], a system designed to align tabular data with knowledge graphs, bridging the gap between structured and unstructured data. By 2021, the challenge saw

refinements in previous systems, with "DAGOBAH" [10] being optimized for more efficient semantic annotations, and "MantisTable V" [11] offering a novel approach to table interpretation.

Systems like "s-elBat" by Cremaschi et al. [12] further explored the challenges of interpreting real-world, messy datadata. The 2022 edition of the challenge introduced specialized datasets such as "SOTAB" [13] and "MammoTab" [14], which closely aligned with the 2023 tasks focusing on Schema.org annotations. In the SemTab 2023 challenge, we introduced DREIFLUSS, a minimalist approach for table matching that leveraged machine learning techniques to perform CTA and CPA tasks using knowledge graphs such as Schema.org and DBpedia [2]. While this approach was effective, it operated within the constraints of a limited number of labels, with Schema.org and DBpedia offering a label set ranging from 46 to 105. For the SemTab 2024 challenge, we have shifted our focus towards the CEA and CTA tasks using the much larger and more semantically rich Wikidata knowledge graph. Given Wikidata's vast label set and comprehensive coverage, we developed a new approach to tackle these tasks using proper techniques leveraging Wikidata API.

3. Tasks

The second round of the SemTab challenge more specifically Accuracy Track emphasizes many tasks out of those we are focusing on two core tasks: CEA and CTA. These tasks aim to enhance table comprehension by assigning specific labels to cells and columns, respectively.

3.1. Cell Entity Annotation (CEA)

CEA involves linking cell values to specific entities from a knowledge base, such as people, places, or organizations. This process enriches the semantic understanding of the table's content, improving data retrieval, integration, and knowledge discovery. Properly annotating cells with relevant entities is crucial for tables with ambiguous or abbreviated terms, which could otherwise lead to misinterpretation. CEA enhances the quality and utility of structured data by ensuring that each cell is connected to a contextually accurate entity.

3.2. Column Type Annotation (CTA)

CTA focuses on categorizing columns by associating them with specific semantic labels that describe their content or purpose. This process involves attributing appropriate labels to columns based on their content, using labels derived from knowledge graphs such as DBpedia and Schema.org. CTA facilitates efficient data integration and enables downstream applications to understand table structure and semantics, proving essential for tasks like data cleaning, schema matching, and query optimization. By providing insights into each column's intended purpose, CTA improves data understanding and analysis.

Together, CEA and CTA tasks aim to enhance table matching and comprehension. These tasks add semantic richness to tables, aiding in data integration, knowledge discovery, and other applications. The following sections will explore the datasets used for CEA and CTA, the experimental setup, the results obtained, and the effectiveness of our approach in addressing these tasks within the SemTab challenge.

4. Dataset

The SemTab 2024 competition[2] features three distinct challenge tracks, with our focus on the Accuracy track. Within this track, various datasets are provided, including WikidataTables2024R1(R2), tBiodiv(L), and tBiomed(L), each consisting of two rounds. Our experiments specifically target the datasets from Round 2, namely tBiodivL[3] and tBiomedL[4], both of which are publicly available on Zenodo. Having these large datasets our approach shows the feasibility in the scalability aspect of it.

For our study, we focused on the CEA and CTA tasks using these datasets. Each dataset is organized into two main subdirectories: *entity* and *horizontal*. Our experiments were conducted using the *horizontal* subdirectory, which is further divided into three subfolders: *gt* (ground truth), *tables*, and *targets*. The *gt* folder contains the ground truth annotations, the *tables* folder includes all possible ground truth annotations for the tables, and the *targets* folder lists all the targets requiring annotation (those without existing ground truth).

Both the biodiversity and biomedical datasets are provided in CSV format. For the Round 2 CEA and CTA tasks, the tBiomedL dataset includes 5,496 tables, while the tBiodivL dataset contains 1,616 tables. The target datasets, also in CSV format, were utilized for evaluation purposes.

5. Methodology

To address the CEA and CTA tasks, we followed a detailed pipeline. The complete implementation, including code, is available on our GitHub repository [5].

5.1. CEA

For the CEA task, we began by loading the CSV file into a DataFrame to streamline processing. The dataset includes three columns: the table name, column index, and row index. To perform the annotation, specific cells were extracted from the table using the provided column and row indices, and these cell values were incorporated into the DataFrame. These values may vary, encompassing strings, paragraphs, and numeric data. Given that some cells contain multiple values, we decided to use only the first value from each cell to simplify the annotation process and mitigate potential ambiguities. The updated DataFrame, as shown in Table 1, reflects these adjustments.

The CEA process aims to link cell values from tabular data to corresponding entities in the Wikidata knowledge graph. This involves assigning a unique Wikidata Entity URI to each cell value, thereby enhancing the semantic enrichment and interoperability of the data.

The methodology for CEA includes the following steps:

[2] https://sem-tab-challenge.github.io/2024/
[3] https://zenodo.org/records/10283083
[4] https://zenodo.org/records/10283119
[5] https://github.com/DKEPassau/CEACTA24

1. **Data Loading and Preparation:** The CSV file was imported into a DataFrame with columns for the table name, column index, and row index. Cells were extracted based on these indices and added to the DataFrame.
2. **Handling Multiple Values:** Since some cells contained more than one value, we opted to use only the first value from each cell to streamline the annotation process.

5.1.1. Rate Limiting and Caching

To adhere to the Wikidata API's rate limits, a `RateLimiter` class was created. This class ensures that API requests do not exceed the maximum allowed frequency, preventing throttling or denial of service. The rate limiter monitors recent API call timestamps and calculates the necessary wait time before making additional requests.

A caching mechanism was also employed using a Python `defaultdict` to store results from previous queries. This approach minimizes redundant API calls, thereby enhancing the overall efficiency of the annotation process.

5.1.2. Entity Identification and URI Construction

To identify the corresponding Wikidata entity for each cell value, we defined the function `get_wikidata_id(category_label)`. This function performs the following steps:

1. Checks if the entity ID for the given category label is available in the cache. If found, it returns the cached ID.
2. If the entity ID is not cached, it invokes the rate limiter's `wait()` method to comply with API rate limits.
3. Sends a GET request to the Wikidata API using the `requests` library with the appropriate search parameters, including the action type, format, language, and label.
4. If the response status is 200 (OK), it parses the JSON response to extract the entity ID. A valid ID is cached and returned; if not found or if the response is malformed, appropriate error messages are logged.

Upon obtaining a valid Wikidata ID, the `construct_entity_uri(wikidata_id)` function constructs the corresponding Wikidata Entity URI.

5.1.3. Processing and Annotation of Tabular Data

The primary function for annotating tabular data is `fetch_and_assign_wikidata_uri(category_label)`, which integrates the above steps to fetch and assign the Wikidata URI for each cell value. This function ensures that each value is a string, removes any leading or trailing whitespace, and then uses `get_wikidata_id` to retrieve the entity ID. If a valid ID is found, the corresponding URI is constructed; otherwise, `None` is returned.

To efficiently apply this function across the dataset, the `process_row(row)` function processes each row of the DataFrame. The `parallel_apply(df, func, workers)` function employs the

`ThreadPoolExecutor` from Python's `concurrent.futures` module to enable parallel processing. This parallelization accelerates the annotation process by distributing the workload across multiple threads. The `parallel_apply` function was configured to use up to 20 worker threads to balance performance and resource utilization.

Finally, the annotated DataFrame, `annotated_target_df`, is produced by applying the `process_row` function to the input dataset (`table_biodiv_cea_target`) using parallel execution.

Table 1
Target DataFrame for CEA after Adding Cell Values

Table Name	Column Index	Row Index	Cell Values	First Cell Value
EGN060702I0010	1	0	Marchamp, Kinly	Marchamp
EGN060702I0010	1	1	Saint-Maurice -de-Gourdans,...	Saint-Maurice -de-Gourdans
EGN060702I0010	1	2	Nivigne et Suran, ...	Nivigne et Suran

5.2. CTA

The CTA process enhances the semantic understanding of dataset columns by mapping them to appropriate types or classes in the Wikidata knowledge graph.

For the CTA task, we started with a CSV file containing two columns: the first specifying the table name and the second providing the column index within the table. This file was loaded into a DataFrame for further processing. An example of the target dataset is shown in Table 2.

Table 2
Example of the CTA Target Dataset

Table Name	Column Index
EGN060702I0010	1
EGN060702I0010	3
EGN060702I0031	1

To perform the annotation, we extracted the specified columns from the indicated tables using the provided column indices. These columns were added to the DataFrame under a new column header, `clean_column_values`. The values in this column were cleaned to retain only unique entries, with multiple values separated by the delimiter "`||`". An example of the cleaned DataFrame is shown in Table 3.

5.2.1. Caching and Rate Limiting

To optimize performance and avoid excessive requests, a local cache (`wikidata_cache`) was implemented. This cache consists of two components: `id_cache` for storing label-to-ID map-

Table 3
Example of the DataFrame After Fetching and Cleaning Column Values

Table Name	Column Index	clean_column_values
EGN060702I0010	1	Marchamp || Saint-Maurice-de-Gourdans
EGN060702I0031	1	Category:Judiciary of Iran || Category:Judiciary of Ukraine
EGN060702I0072	2	Wikipedia:Vital articles/Level/4 || Wikipedia:Vital articles/Level/5

pings and `related_cache` for storing related entity IDs. A rate-limiting decorator was applied to ensure that no more than 10 requests per second are made, adhering to Wikidata's API rate limits and improving overall efficiency.

5.2.2. Entity Identification and Relation Mapping

The function `get_wikidata_id` is used to retrieve the Wikidata ID for each label in the `clean_column_values`. If the ID is not already present in the cache, the function sends a request to the Wikidata API and updates the cache with the result. Additionally, the function `get_related_ids` retrieves related IDs based on properties such as `P31` (instance of) and `P279` (subclass of), which are crucial for determining the semantic type or class of the column values.

5.2.3. Processing and Annotation of Columns

The `process_cell` function processes each entry in the `clean_column_values` column. This function splits the values, filters out irrelevant entries, and deduplicates them. For each unique label, it retrieves the Wikidata ID and associated subclass IDs. These subclass IDs are then aggregated, and the most frequently occurring ones are selected as the final column type annotation.

5.2.4. Cache Management

To maintain efficiency and reduce redundant API requests, the cache is saved to a file at the end of the script execution using the `save_cache` function. When the script is restarted, the `load_cache` function reloads the cache, preserving previously obtained results and ensuring more efficient subsequent executions.

In summary, the CTA process involves extracting, cleaning, and annotating column data using the Wikidata knowledge graph, with caching and rate limiting employed to optimize performance and resource utilization.

6. Results

We evaluated the performance of our methodology by applying it to the CEA and CTA tasks on datasets such as tBiodivL and tBiomedL. This evaluation utilized the target datasets provided by

the SemTab organizers[6]. Our results underscore the effectiveness of our approach, particularly regarding F1 and Precision scores.

For the SemTab 2024 competition, we focused on two primary datasets: tBiodiv-Large-Relational and tBiomed-Large-Relational. Our methodology demonstrated strong performance, achieving F1 scores between 61% and 93% across both CTA and CEA tasks. These results are summarized in Table 4.

Table 4
Precision, recall, and F1 scores for CEA and CTA tasks on tBiodiv-Large-Relational and tBiomed-Large-Relational datasets.

Dataset	Task	F1 Score	Precision
tBiodiv-Large-Relational	CEA	93.20%	93.20%
tBiodiv-Large-Relational	CTA	61.50%	61.50%
tBiomed-Large-Relational	CEA	92.50%	92.50%

7. Discussion

Our results demonstrate the effectiveness of our proposed methodology for CEA and CTA on the SemTab 2024 datasets. The methodology utilized pre-existing semantic resources from Wikidata to enhance table annotation tasks, showcasing significant improvements in both accuracy and efficiency.

7.1. Performance Insights

The CEA task achieved impressive F1 scores, reaching up to 93.20% for the tBiodiv-Large-Relational dataset and 92.50% for the tBiomed-Large-Relational dataset, indicating high precision in linking cell values to Wikidata entities. These high scores reflect the robustness of our system in identifying and annotating cell values accurately, which is crucial for integrating and enriching tabular data with semantic information.

In contrast, the CTA task showed a broader range of F1 scores, with the tBiodiv-Large-Relational dataset reaching 61.50%. While this score is lower compared to CEA, it still represents a significant achievement in classifying column types. The variability in CTA performance could be attributed to the complexity and diversity of column types across different datasets, which may affect the consistency of the annotations.

7.2. Methodological Contributions

Our approach leverages the rich semantic labels provided by Wikidata, enhancing the accuracy of table annotations by providing standardized and comprehensive semantic details. The integration of these labels allows for more precise and meaningful annotations, which improve the interoperability and usability of the annotated data.

[6]https://sem-tab-challenge.github.io/2024/results.html

The implementation of rate limiting and caching mechanisms has proven essential in managing API usage and optimizing performance. By reducing redundant API requests and adhering to rate limits, our system efficiently handles large-scale data processing, which is critical for real-world applications involving extensive datasets.

7.3. Future Work

Future research could focus on integrating additional knowledge graphs or domain-specific ontologies to overcome the limitations of relying solely on Wikidata. Enhancing the performance of the CTA task may benefit from the development of more advanced classification models or the inclusion of richer features from the datasets. Expanding the methodology to accommodate multilingual and domain-specific datasets could further broaden its applicability across diverse contexts and industries. Additionally, the current approach will be extended into a more comprehensive framework based on our previous work [2], allowing for scalability and the potential incorporation of machine learning techniques.

In conclusion, our methodology presents a sound approach in the field of table annotation, offering a scalable and effective approach to integrating semantic information into tabular data. The positive results achieved in both CEA and CTA tasks demonstrate the potential of combining pre-existing semantic resources with innovative processing techniques to enhance data interoperability and knowledge discovery.

References

[1] A. O. Shigarov, Table understanding: Problem overview, WIREs Data Mining Knowl. Discov. 13 (2023). URL: https://doi.org/10.1002/widm.1482.

[2] V. R. Parmar, A. Algergawy, DREIFLUSS: A minimalist approach for table matching, in: V. Efthymiou, E. Jiménez-Ruiz, J. Chen, V. Cutrona, O. Hassanzadeh, J. Sequeda, K. Srinivas, N. Abdelmageed, M. Hulsebos, A. Khatiwada, K. Korini, B. Kruit (Eds.), Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching, SemTab 2023, co-located with the 22nd International Semantic Web Conference, ISWC 2023, Athens, Greece, November 6-10, 2023, volume 3557 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 50–60. URL: https://ceur-ws.org/Vol-3557/paper4.pdf.

[3] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann, Dbpedia - A crystallization point for the web of data, J. Web Semant. 7 (2009) 154–165. URL: https://doi.org/10.1016/j.websem.2009.07.002.

[4] R. V. Guha, D. Brickley, S. Macbeth, Schema.org: Evolution of structured data on the web, ACM Queue 13 (2015) 10. URL: https://doi.org/10.1145/2857274.2857276.

[5] D. Oliveira, M. d'Aquin, ADOG - annotating data with ontologies and graphs, volume 2553 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 1–6. URL: https://ceur-ws.org/Vol-2553/paper1.pdf.

[6] M. Cremaschi, R. Avogadro, D. Chieregato, Mantistable: an automatic approach for the semantic table interpretation, volume 2553 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 15–24. URL: https://ceur-ws.org/Vol-2553/paper3.pdf.

[7] A. Thawani, M. Hu, E. Hu, H. Zafar, N. T. Divvala, A. Singh, E. Qasemi, P. A. Szekely, J. Pujara, Entity linking to knowledge graphs to infer column types and properties, volume 2553 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 25–32. URL: https://ceur-ws.org/Vol-2553/paper4.pdf.

[8] V. Huynh, J. Liu, Y. Chabot, T. Labbé, P. Monnin, R. Troncy, DAGOBAH: enhanced scoring algorithms for scalable annotations of tabular data, volume 2775 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 27–39. URL: https://ceur-ws.org/Vol-2775/paper3.pdf.

[9] N. Abdelmageed, S. Schindler, Jentab: Matching tabular data to knowledge graphs, volume 2775 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 40–49. URL: https://ceur-ws.org/Vol-2775/paper4.pdf.

[10] V. Huynh, J. Liu, Y. Chabot, F. Deuzé, T. Labbé, P. Monnin, R. Troncy, DAGOBAH: table and graph contexts for efficient semantic annotation of tabular data, volume 3103 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 19–31. URL: https://ceur-ws.org/Vol-3103/paper2.pdf.

[11] R. Avogadro, M. Cremaschi, Mantistable V: A novel and efficient approach to semantic table interpretation, volume 3103 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 79–91. URL: https://ceur-ws.org/Vol-3103/paper7.pdf.

[12] M. Cremaschi, R. Avogadro, D. Chieregato, s-elbat: A semantic interpretation approach for messy table-s, volume 3320 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 59–71. URL: https://ceur-ws.org/Vol-3320/paper7.pdf.

[13] K. Korini, R. Peeters, C. Bizer, SOTAB: the WDC schema.org table annotation benchmark, volume 3320 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 14–19. URL: https://ceur-ws.org/Vol-3320/paper1.pdf.

[14] M. Marzocchi, M. Cremaschi, R. Pozzi, R. Avogadro, M. Palmonari, Mammotab: A giant and comprehensive dataset for semantic table interpretation, volume 3320 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 28–33. URL: https://ceur-ws.org/Vol-3320/paper3.pdf.