CitySTI 2024 System: Tabular Data to KG Matching using LLMs

Dylan Li Tin Yue[1], Ernesto Jimenez-Ruiz[1,*]

[1]City St George's, University of London

Abstract

This paper investigates the use of a Large Language Model (LLM) to match tabular data with knowledge graphs. The system participated in the STI vs. LLMs 2024 SemTab Track, which prompts a model to perform the cell entity annotation (CEA) task. The study covers the processes from data cleaning and matching to its execution in the cloud, while relying on a Lookup API to generate a list of candidates. This project not only contributes to the understanding of the applications of Large Language Models in tabular data annotations but also lays the groundwork for future research in the field.

Keywords

Tabular Data Annotation, Knowledge Graphs, Large Language Model, SemTab Challenge, Entity Matching

1. Introduction

Over the past few decades, the use of data has increased dramatically where a large portion of this data is structured as tabular data. This increase is largely because of the growing trend of Open Data publication, which makes data increasingly accessible to the public.

The range of knowledge extracted from the data has not grown in proportion to the exponential increase in data volume. This is mostly because tools and expertise needed are limited in this area of work where datasets are usually large and complex. Tabular data is widely available and accessible on the internet, data silos and data lakes. These tables are crucial as they are hugely demanded in activities such as data analytics, data mining and data integration tasks.

However, this form of data often lacks the contextual understanding that is required for the users and machines to properly interpret. To fully exploit its potential, it is necessary to understand its semantic structure and underlying meaning. Even though Knowledge Graphs [1] address this issue by providing insights into the significance of the data, the process is still mostly manual.

With the rise of ChatGPT, Large Language Models (LLMs) gained in popularity and are increasingly changing important aspects of our lives. This type of Artificial Intelligence (AI) is currently trained in the order of trillions of tokens. As a result, it appears to recognize and understand human text and can act as a large source of (parametric) knowledge. Their value and benefits are being widely acknowledged in multiple tasks, which leads to the following research question:

"To what extent can LLMs be applied in the context of matching tabular data to knowledge graph?"

SemTab Challenge. Tabular format is one of the most popular ways for organizations to store data. Tabular data to Knowledge Graph (KG) matching is the process to map elements of a table data to its corresponding semantic tags within a KG such as Wikidata [2] and DBpedia [3]. The SemTab challenge[1] [4, 5] has contributed to the systematic evaluation of systems tackling the above task, also known as Semantic Table Interpretation (STI) systems.

Tables are often of poor data quality because of incomplete or missing metadata. Understanding the semantic meaning and context might therefore be difficult when metadata such as proper table titles,

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

*Corresponding author.

ernesto.jimenez-ruiz@city.ac.uk (E. Jimenez-Ruiz)

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

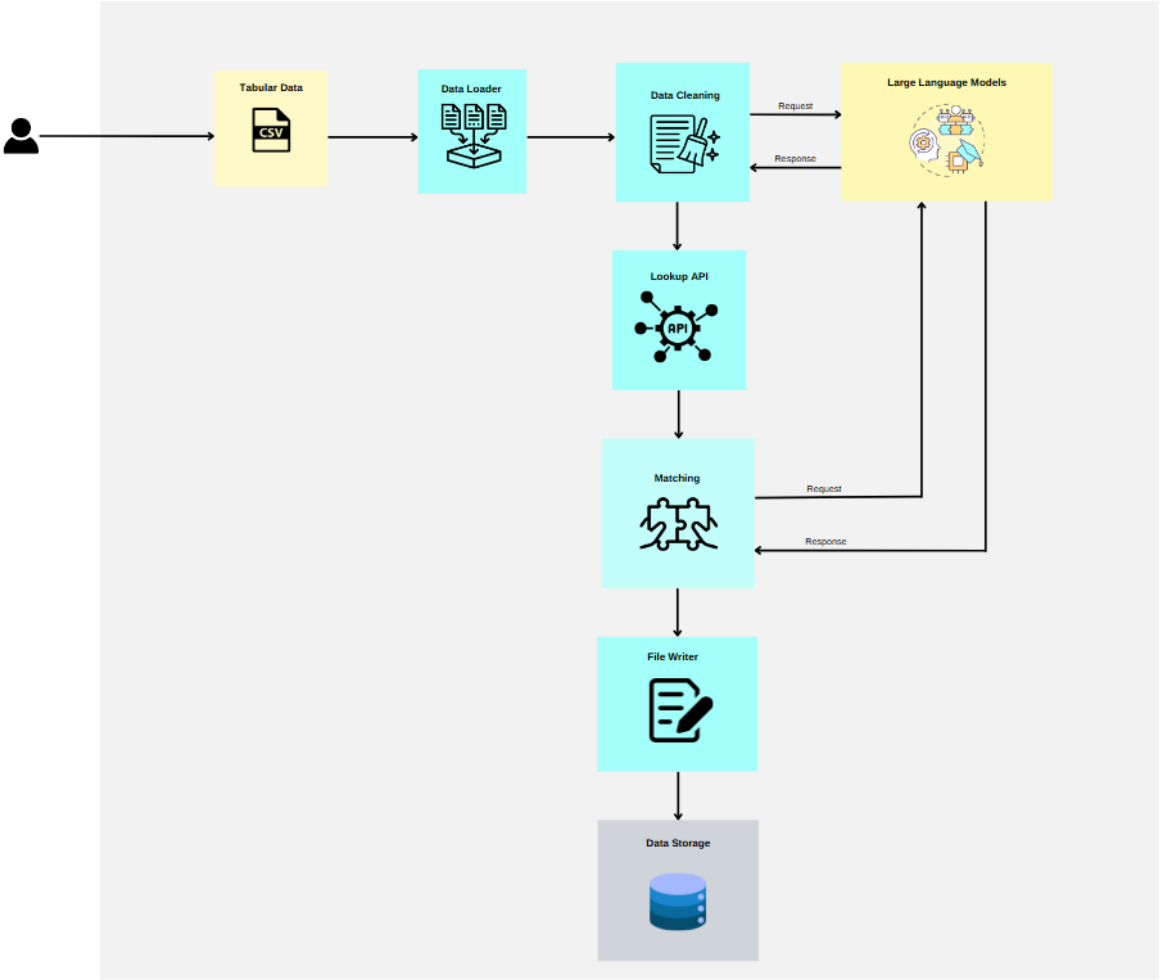

Figure 1: Architecture Diagram

relationships between the different data elements or column names are unavailable. Another challenge is that values in cells can be noisy, contain typos, abbreviations and ambiguous names that cannot be matched with the KG. Along with the data quality issues, other challenges can arise within the semantic matching process since relationships between columns are not known and the table columns can represent a more specific or general concept. Moreover, as knowledge is constantly and quickly evolving in the current world, KGs may not always have the most up-to-date information needed.

We have targeted the *STI vs LLMs* SemTab 2024 track, which involves the use of only LLMs to perform the CEA task which is to match a data cell to a KG entity. Round 1 of the challenge uses the *SuperSemTab 24* dataset which has been Automatically Generated (AG). The test set of this dataset includes 74,837 cells to be annotated from 4,044 CSV files, each containing an average of five rows of data. The dataset used for Round 2 is *MammoTab 24* [6] which was extracted from 21,149,260 Wikipedia pages. Unlike Round 1, it consists of 2,500 tables for the training set and 500 tables for the testing set. Both SuperSemTab and MammoTab target Wikidata as KG.

Related Work. Based on a recent survey carried out on semantic interpretation of tabular data [7], more than 85 systems have been proposed to tackle the tabular data to KG matching problem since 2007. Recently there has been an increase of approaches relying on the features of pre-trained and large language models like Do-duo [8], TURL [9], DAGOBAH SL 2022 [10], TorchicTab [11], Korini and Bizer [12], TableGPT [13], and TableLlama [14]. In the SemTab 2024 challenge [5], only three systems have participated in the *STI vs. LLMs* track: `CitySTI` (ours), TSOTSA [15], and Kepler-aSI [16].

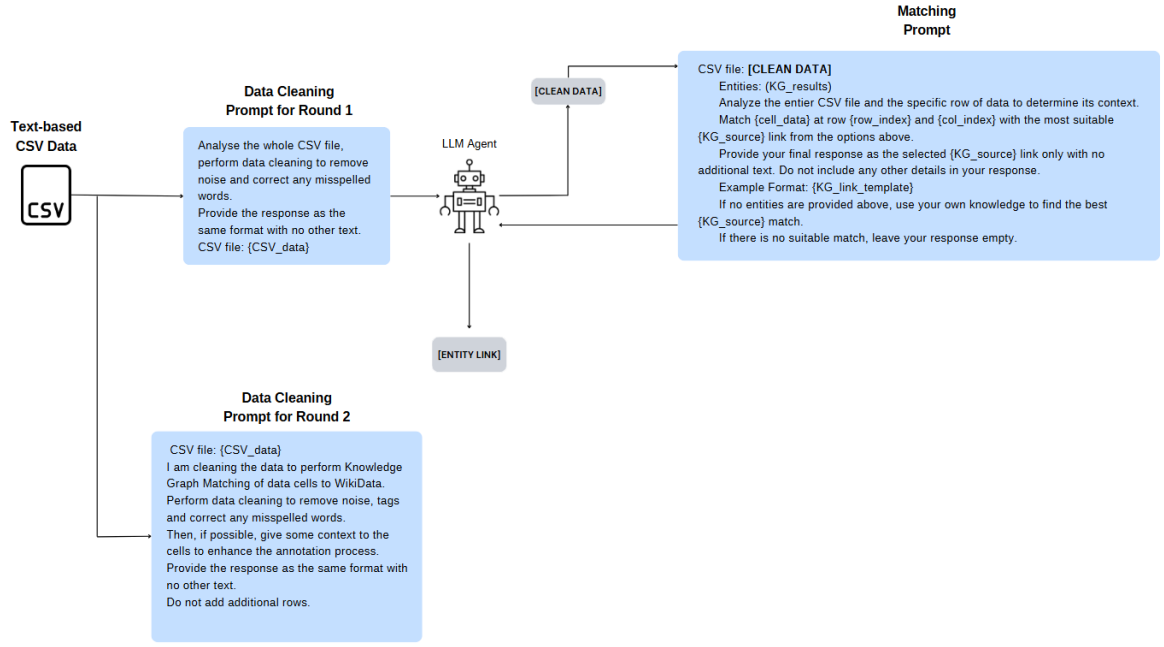

Figure 2: Entity Matching using LLM Workflow Diagram

United Kingdom	id: http://www.wikidata.org/entity/Q145, label: United Kingdom, description: country in north-west Europe
	id: http://www.wikidata.org/entity/Q174193, label: United Kingdom of Great Britain and Ireland, description: historical sovereign state (1801–1922)
	id: http://www.wikidata.org/entity/Q161885, label: Kingdom of Great Britain, description: constitutional monarchy in Western Europe (1707–1800)
	id: http://www.wikidata.org/entity/Q11010, label: Parliament of the United Kingdom, description: supreme legislative body of the United Kingdom
	id: http://www.wikidata.org/entity/Q7979, label: British English, description: forms of the English language used in Britain
Elvis Presley	id: http://www.wikidata.org/entity/Q303, label: Elvis Presley, description: American singer and actor (1935–1977)
	id: http://www.wikidata.org/entity/Q610926, label: Elvis Presley, description: 1956 self-titled debut studio album by Elvis Presley
	id: http://www.wikidata.org/entity/Q83488456, label: Elvis Presley, description: 1956 US LP by Elvis Presley; RCA Victor – LPM-1254
	id: http://www.wikidata.org/entity/Q4655458, label: A Big Hunk o' Love, description: 1959 single by Elvis Presley
	id: http://www.wikidata.org/entity/Q47509719, label: Elvis Presley, description: painting by Ralph Wolfe Cowan

Figure 3: Example of Wikidata candidates.

2. The CitySTI System

CitySTI combines state-of-the-art LLMs with natural language processing (NLP) techniques to perform the SemTab's CEA task. It does not only handle the data matching with the appropriate entities but also performs data cleaning. Figure 1 provides a general overview of the different components of CitySTI, which are summarised as follows.

Data cleaning. This component aims at cleaning the input tabular data from noise and correcting any misspelled words. Figure 2 shows the used prompts to guide the LLM in the process. Note that we used different prompts in each of the rounds of the *STI vs LLMs* SemTab 2024 track.

Candidate generator (lookup). The candidates are extracted via the lookup service provided by the target KG. Given an input query (*e.g.*, the text value of a cell) the look-up service extracts candidates (partially) matching the query. CitySTI implements an API to access the lookup services of different KGs and extract the top-5 candidates for each query (see Figure 3).

Matching. This component associates the data cell with its appropriate KG entity. It gets as input the top-5 candidates from the lookup component and communicates with an LLM model to identify the best choice. The set of candidates are fed within the matching prompt along with the clean CSV data to perform the matching (see Figure 2). If only one candidate is retrieved from the lookup, it will be

automatically assigned to its cell data without any need of a matching prompt. An entity cache was also implemented as values to be annotated usually have frequent occurrences. It uses a dictionary that stores the pools of candidates to avoid repeated retrieval of the KG entity for the same cell data in the same table. The cache was automatically cleared every 15 seconds to prevent memory overload.

The matching component implemented two different approaches, tailored to the specific dataset in each of the rounds of the *STI vs LLMs* SemTab 2024 track, as described next. Codes for both approaches are available in this GitHub repository: https://github.com/dylanlty/CitySTI-2024

LLM component - approach round 1. In this approach, the system loops through all the CSV files in a folder, and annotates each cell data, while skipping the numeric and date values. The system then feeds the entire table to the *GPT-4o-mini* model to improve its relevance and accuracy when performing the data cleaning and matching.

All the components were run on a Virtual Machine (VM) instance in Google Cloud Platform's Compute Engine. Google Cloud Platform was used to process the large volumes of data. This was essential because matching the large volumes of data cells to a KG is very time and resources demanding. Before creating a VM instance through the Compute Engine component, it was properly set up to avoid any performance bottleneck. Since a large amount of memory may be required (*e.g.*, to store the entities in the dictionary), 8 GB of memory was chosen. The N2 machine series was selected to run as this works well for the kind of task required where the system only involved annotating data cells and performing read-write operations. Additionally, a 10 GB SSD was allocated to store all the input data and the necessary packages.

During the implementation of the system, open-source models on HuggingFace like the Llama-3-8B-Instruct were explored and then run on *RunPod*.[2] The performance of these models did not match as those currently provided by Google's Gemini or OpenAI's GPT models. It proved to be costly to operate and significantly more expensive than the APIs offered by the other major AI providers (see Section 3).

LLM component - approach round 2. The second round of the challenge undertook some changes to address some issues faced in the previous round and because the MammoTab 24 dataset is more challenging in terms of the size of the tables. The first change was that the system now reads and annotates based on the target file (*i.e.*, cells for which there is ground truth) given by the SemTab challenge instead of processing all potential cells.

Another different approach taken involved processing and providing context in batches instead of the entire tabular data and the use of the *Gemini-1.5-flash* model. This was done to address the token usage limits set by the LLM Provider. Unlike the approach in Round 1, the datasets are loaded into a Pandas DataFrame, cleaned in batches of 15 rows at a time, and written to a temporary file. This file is then sliced into different rows to provide the model some context when matching. The number of rows provided varies from 4 to 6, depending on the position of the data cell.

3. Results

Table 1 shows the comparison of the three tested LLM models. Relying on the public dataset of the round 1 dataset of the SuperSemtab 24 track, GPT-4o-mini produced the best results in terms of F1 Score, closely followed by Gemini-1.5-flash. Gemini-1.5-flash was time and cost-effective in comparison with the other two models.

As discussed in the previous section, GPT-4o-mini was the choice for round 1, while Gemini-1.5-flash was the model for round 2. Table 2 shows the official results reported in the SemTab 2024 challenge for the *STI vs LLMs* track. `CitySTI` produces competitive results with respect to the other two participants (TSOTSA and Kepler-aSI).

[2]RunPod: https://www.runpod.io/

LLM	Time (h)	Cost ($)	F1 Score	Precision	Recall
Gemini-1.5-flash	**4.30**	**1.93**	0.861	**0.889**	0.834
Llama-3-8B-Instruct	14.86	21.28	0.763	0.774	0.752
GPT-4o-mini	35.80	2.43	**0.869**	0.879	**0.860**

Table 1
Comparison of LLMs on Round 1 (on the provided training dataset).

Round	CitySTI		TSOTSA		Kepler-aSI	
	F1 Score	Precision	F1 Score	Precision	F1 Score	Precision
Round 1 - SuperSemtab 24	0.858	0.866	**0.905**	0.905	-	-
Round 2 - MammoTab 24	**0.647**	0.648	-	-	0.182	0.336

Table 2
Official SemTab 2024 results (on test data) for the *STI vs LLMs* track [5].

4. Conclusions, Challenges, and Future Work

This paper presented `CitySTI`, a system to perform the annotations of tabular data to KG. The approach taken to tackle the tasks makes use of prompting an LLM model to clean the data and match the KG entities. The results of it are satisfactory considering it is the first time delving into the SemTab challenge together with the realm of LLMs. Since `CitySTI` relies only on prompting, it can be further improved and may even have a greater performance if fine-tuning is implemented. This will remove the need for large prompts to achieve the desirable output and reduce the number of tokens needed. The prompt structure can be further optimized to enable the system to process several data cells in a single prompt which would significantly reduce the number of requests being sent. Another aspect of the development of the system that can be improved is its ability to be more fault tolerance in terms that it can recover from errors. It should be able to continue where it left on and this would save significant amount of time as opposed to manually removing or adding the files.

Limitations. While many LLMs are free and open-sourced, OpenAI and Google charge for their APIs when using one of their models. The main reason behind this is they have to maintain and operate these advanced AI models and since they are expensive to run, these models come with usage restrictions that add extra challenges to the task. The usage restrictions can come in the form of rate or token limits. Even though Google provides a free-tier option for its Gemini API, the rate limit is insufficient to annotate several thousands of cell data.

Additionally, the lookup search for entities on Wikidata is not very flexible as it is unable to recognize typos or incorporate context words related to the searched term like a modern search engine does. The accuracy could have been significantly improved if the lookup search was more advanced. In the near future, we plan to explore alternative APIs to access the target KG.

References

[1] G. Weikum, Knowledge Graphs 2021: A Data Odyssey, Proc. VLDB Endow. 14 (2021) 3233–3238. URL: http://www.vldb.org/pvldb/vol14/p3233-weikum.pdf.

[2] D. Vrandecic, M. Krötzsch, Wikidata: a free collaborative knowledge base, Commun. ACM 57 (2014) 78–85.

[3] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: A Nucleus for a Web of Open Data, in: The Semantic Web, Springer Berlin Heidelberg, 2007, pp. 722–735.

[4] E. Jimenez-Ruiz, O. Hassanzadeh, V. Efthymiou, J. Chen, K. Srinivas, SemTab 2019: Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems, in: The Semantic Web: ESWC, Springer International Publishing, 2020.

[5] 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, 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.

[6] 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.

[7] M. Cremaschi, B. Spahiu, M. Palmonari, E. Jimenez-Ruiz, Survey on Semantic Interpretation of Tabular Data: Challenges and Directions, arXiv preprint arXiv:2411.11891 (2024).

[8] Y. Suhara, J. Li, Y. Li, D. Zhang, c. Demiralp, C. Chen, W.-C. Tan, Annotating columns with pre-trained language models, in: Proceedings of the 2022 International Conference on Management of Data, SIGMOD '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 1493–1503. URL: https://doi.org/10.1145/3514221.3517906. doi:10.1145/3514221.3517906.

[9] X. Deng, H. Sun, A. Lees, Y. Wu, C. Yu, TURL: table understanding through representation learning, Proc. VLDB Endow. 14 (2020) 307–319. URL: http://www.vldb.org/pvldb/vol14/p307-deng.pdf. doi:10.5555/3430915.3442430.

[10] V.-P. Huynh, Y. Chabot, T. Labbé, J. Liu, R. Troncy., From Heuristics to Language Models: A Journey Through the Universe of Semantic Table Interpretation with DAGOBAH, in: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab), CEUR-WS.org, 2022.

[11] I. Dasoulas, D. Yang, X. Duan, A. Dimou, TorchicTab: Semantic Table Annotation with Wikidata and Language Models, in: SemTab'23: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching 2023, co-located with the 22nd International Semantic Web Conference (ISWC), 2023.

[12] K. Korini, C. Bizer, Column Type Annotation using ChatGPT, 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, 2023. URL: https://ceur-ws.org/Vol-3462/TADA1.pdf.

[13] L. Zha, J. Zhou, L. Li, R. Wang, Q. Huang, S. Yang, J. Yuan, C. Su, X. Li, A. Su, T. Zhang, C. Zhou, K. Shou, M. Wang, W. Zhu, G. Lu, C. Ye, Y. Ye, W. Ye, Y. Zhang, X. Deng, J. Xu, H. Wang, G. Chen, J. Zhao, Tablegpt: Towards unifying tables, nature language and commands into one GPT, CoRR abs/2307.08674 (2023). URL: https://doi.org/10.48550/arXiv.2307.08674. doi:10.48550/ARXIV.2307.08674. arXiv:2307.08674.

[14] T. Zhang, X. Yue, Y. Li, H. Sun, Tablellama: Towards open large generalist models for tables, CoRR abs/2311.09206 (2023). URL: https://doi.org/10.48550/arXiv.2311.09206. doi:10.48550/ARXIV.2311.09206. arXiv:2311.09206.

[15] 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.

[16] 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.